



What's new

- ▶ [View information on object dependencies](#)
- ▶ [Error checking in forms and reports](#)
- ▶ [Propagating field properties](#)
- ▶ [Smart tags](#)
- ▶ [Back up a database or project](#)
- ▶ [Windows XP theme support](#)
- ▶ [Improved sorting in controls](#)
- ▶ [Autocorrect options](#)
- ▶ [Enhanced font capabilities in SQL views](#)
- ▶ [Context-based Help in SQL view](#)

- ▶ [Importing, exporting, and linking](#)

- ▶ [XML support](#)

- ▶ [Security enhancements](#)

- ▶ [More new features](#)

[Show All](#)

Get started with Access 2003

There are different ways you can get started with Microsoft Office Access 2003, depending on your experience level.

[Upgrading from previous versions of Microsoft Access](#)

You can use a [Microsoft Access 2000 database](#) or [Access 2000 project](#) in both Access 2003 and Access 2000— even if the file is a security-enabled Access database. If you also want to make an Access database available to users of Access 97 or earlier, use one or more of the following procedures.

[Use a one-file Access database with several versions of Microsoft Access](#)

If your [Microsoft Access database](#) is in one file, and you want to use it in several versions of Microsoft Access, convert it to a [front-end/back-end application](#). The data remains in the oldest version of Access, and you use a front end that has been converted to Access 2000 or Access 2003.

1. Convert the Access database to Access 2000 or Access 2002-2003 file format. When converting, specify a new name for the file.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access 2003.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.

7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
2. On the **Tools** menu, point to **Database Utilities** and use the **Database Splitter** to split the converted database into a front-end/back-end application.
3. Delete the back-end database that the **Database Splitter** created.

You want your data to remain in the original database, so you will use the original database as the back-end database. The back-end database should be in the oldest version of Microsoft Access that is being used.

4. On the **Tools** menu, point to **Database Utilities** and use the **Linked Table Manager** to link the new front-end database to the tables in the previous-version database.

You can then enhance the new front-end database to support new features for users that have upgraded to Microsoft Access 2000 or Access 2003. Users of previous versions can continue to use the previous-version database. For example, if the back-end tables are in Microsoft Access version 2.0 format, you can use up to four versions of Microsoft Access: Microsoft Access version 2.0 (using the original database), Microsoft Access 95 and 97 (using an enabled version of the original database or a converted front-end database), and Microsoft Access 2000 or Access 2003 (using a converted front-end database).

 [Use a front-end/back-end application with several versions of Microsoft Access](#)

If your [Microsoft Access database](#) is already a [front-end/back-end application](#), you only need to convert the front end to Access 2000 or Access 2002-2003 file format.

1. Leave the back-end database alone.
2. Convert the front-end database to Access 2000 or Access 2002-2003 file format.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access 2003.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
3. On the **Tools** menu, point to **Database Utilities** and use the **Linked Table Manager** to link the new Access 2000 or Access 2003 front-end database to the tables in the previous-version back-end database. You can then enhance the Access 2000 or Access 2003 front-end database to support new features for users that have upgraded to Access 2000 or Access 2003.

[Open a database created in Access 97 or earlier without converting it](#)

You can open a [Microsoft Access database](#) created in Access 97 or earlier without converting it.

1. Make sure that the Access database you're going to [enable](#) is closed. If the

file is a [multiuser](#) Access database located on a server or in a shared folder, make sure that no one else has it open.

2. Click **Open**  on the **Database** toolbar.
3. In the **Open** dialog box, click the previous-version Access database you want to enable and click **Open**.
4. In the **Convert/Open Database** dialog box, click **Open Database**.

Microsoft Access [enables](#) the previous-version database by storing additional information in the file to accommodate the default file format specified on the **Advanced** tab of the **Options** dialog box. An enabled database can still be opened with its original version of Microsoft Access.

 [Use a security-enabled Access database with multiple versions of Access](#)

With one exception, the issues involved when sharing a security-enabled database across more than one version of Microsoft Access are the same as the issues for sharing an unsecured database across more than one version. The one exception concerns how to handle the [workgroup information files](#) that are used with the security-enabled database.

- Do one of the following:
 - If you want to share a security-enabled database from Access 95 or 97, tell users who will be upgrading to Access 2003 to join the appropriate workgroup information file with the oldest version of Access that will be sharing the security-enabled database.

 [How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.

3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

Access 2003 can use workgroup information files that have been created with any previous version. With one exception, a previous version of Access can't use a workgroup information that was created in a later version. The exception is Access 2000, which can use a workgroup information file that was created in Access 2003 if that workgroup information file is in Access 2000 file format.

Important If users will be sharing a security-enabled database from Microsoft Access 95 or 97, you should compact the current workgroup information file with Access 2003 before using it. Compacting the file by using Access 2003 does not change the file format, so the file can continue to be used by any Microsoft Access 95 or 97 users who are not upgrading.

- If the shared database is Microsoft Access version 2.0, convert the workgroup information file that will be used with the security-enabled database.

[How?](#)

You do not need to convert a Microsoft Access 2000 [workgroup information file](#) to Access 2003. However, to take advantage of security and performance improvements, you should re-create workgroup information files from Access 97 or earlier as described below.

1. Create a new workgroup information file, making sure to enter the exact, case-sensitive name, company name, and [workgroup](#)

[ID](#) that was used to create the original file. Failure to re-enter the exact entries that were used to create the original file will create an invalid [Admins group](#).

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.
6. Click **OK**.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by

your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

2. Re-create any group accounts, making sure to enter the exact, case-sensitive group name and [personal ID \(PID\)](#) for each group.

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries.

If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

3. Re-create each user account, making sure to enter the exact, case-sensitive user name and PID for each user.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts.

2. Open a database.

3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.

4. On the **Users** tab, click **New**.

5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
- The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

Tell only users who are upgrading to Microsoft Access 2000 or Access 2003 to join the converted workgroup information file. All users who are not upgrading from version 2.0 must continue to use the workgroup information file produced with their original version.

[Already familiar with other database or spreadsheet applications](#)

If you are familiar with other database or spreadsheet applications, then you probably know the basics of how these applications work and what databases are used for. Microsoft Access differs from many other database applications by allowing you to create [relational databases](#). Access also provides many options for working with other database programs such as Microsoft SQL Server.

[Relationships in a database](#)

After you've set up different tables for each subject in your [Microsoft Access database](#), you need a way of telling Microsoft Access how to bring that information back together again. The first step in this process is to define [relationships](#) between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once. For example, this form includes information from four tables:

Product:	Unit Price:	Quantity:
Guinness Stout	\$24.00	14
Scottish Longbread	\$12.50	20
Pâté chinois	\$36.00	10
*		

- 1 The Customers table
- 2 The Orders table
- 3 The Products table
- 4 The Order Details table

[How relationships work](#)

In the previous example, the fields in four tables must be coordinated so that they show information about the same order. This coordination is accomplished with relationships between tables. A relationship works by matching data in key fields—usually a field with the same name in both tables. In most cases, these matching fields are the [primary key](#) from one table, which provides a unique identifier for each record, and a [foreign key](#) in the other table. For example, employees can be associated with orders they're responsible for by creating a relationship between the EmployeeID fields.

Employee ID	Last Name	First Name
1	Davolio	Nancy

Order ID	Customer ID	Employee ID
10022	LAMAI	1

- 1 EmployeeID appears in both tables— -as a primary key ...
- 2 ... and as a foreign key.

[A one-to-many relationship](#)

A one-to-many relationship is the most common type of relationship. In a one-

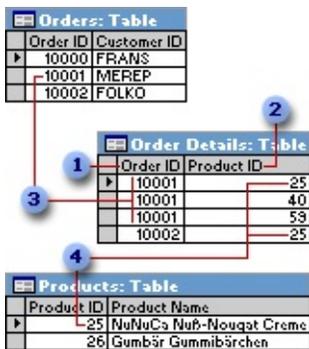
to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.



- 1 One supplier ...
- 2 ... can supply more than one product ...
- 3 ... but each product has only one supplier.

[A many-to-many relationship](#)

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose **primary key** consists of two fields— the **foreign keys** from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table. For example, the Orders table and the Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.



- 1 Primary key from the Orders table

- 2 Primary key from the Products table
- 3 One order can have many products ...
- 4 ... and each product can appear on many orders.

[A one-to-one relationship](#)

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. For example, you might want to create a table to track employees participating in a fundraising soccer game. Each soccer player in the Soccer Players table has one matching record in the Employees table.

Employees: Table		
Employee ID	Last Name	First Name
1	Davolio	Nancy
2	Fuller	Andrew
3	Leverling	Janet
4	Peacock	Margaret
5	Buchanan	Steven

Soccer Players: Table		
Employee ID	Player Nickname	Skill Level
1	Slammin' Man	2
2	Ace	1
5	Stevemeister	2

- 1 Each soccer player has one matching record in the Employees table.
- 2 This set of values is a subset of the EmployeeID field and the Employees table.

[About defining relationships](#)

The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

- A one-to-many relationship is created if only one of the related fields is a [primary key](#) or has a [unique index](#).
- A one-to-one relationship is created if both of the related fields are primary keys or have unique indexes.
- A many-to-many relationship is really two one-to-many relationships with a

third table whose primary key consists of two fields— the [foreign keys](#) from the two other tables.

You can also create a relationship between a table and itself. This is useful in situations where you need to perform a Lookup within the same table. In the Employees table, for example, you can define a relationship between the EmployeeID and ReportsTo fields, so that the ReportsTo field can display employee data from a matching EmployeeID.

Note If you drag a field that isn't a primary key and doesn't have a unique index to another field that isn't a primary key and doesn't have a unique index, an indeterminate relationship is created. In queries containing tables with an indeterminate relationship, Microsoft Access displays a default [join](#) line between the tables, but [referential integrity](#) won't be enforced, and there's no guarantee that records are unique in either table.

[Referential integrity](#)

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. You can set referential integrity when all of the following conditions are met:

- The matching field from the [primary table](#) is a [primary key](#) or has a [unique index](#).
- The related fields have the same [data type](#). There are two exceptions. An [AutoNumber](#) field can be related to a Number field with a **FieldSize** property setting of **Long Integer**, and an AutoNumber field with a **FieldSize** property setting of **Replication ID** can be related to a Number field with a **FieldSize** property setting of **Replication ID**.
- Both tables belong to the same Microsoft Access database. If the tables are linked tables, they must be tables in Microsoft Access format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

The following rules apply when you use referential integrity:

- You can't enter a value in the [foreign key](#) field of the related table that

doesn't exist in the primary key of the primary table. However, you can enter a [Null](#) value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.

- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

[Cascading updates and deletes](#)

For relationships in which [referential integrity](#) is enforced, you can specify whether you want Microsoft Access to automatically [cascade update](#) and [cascade delete](#) related records. If you set these options, delete and update operations that would normally be prevented by referential integrity rules are allowed. When you delete records or change [primary key](#) values in a [primary table](#), Microsoft Access makes necessary changes to related tables to preserve referential integrity.

If you select the **Cascade Update Related Fields** check box when defining a relationship, any time you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship isn't broken. Microsoft Access cascades updates without displaying any message.

Note If the primary key in the primary table is an [AutoNumber](#) field, setting the **Cascade Update Related Fields** check box will have no effect, because you can't change the value in an AutoNumber field.

If you select the **Cascade Delete Related Records** check box when defining a relationship, any time you delete records in the primary table, Microsoft Access automatically deletes related records in the related table. For example, if you

delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the **Cascade Delete Related Records** check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a [delete query](#), Microsoft Access automatically deletes the records in related tables without displaying a warning.

[Working with other applications](#)

Microsoft Access 2003 provides features for working with other programs.

- **Work with Microsoft SQL Server** You can create a [Microsoft Access project](#) that is easy to connect to a Microsoft SQL Server database, or use the Microsoft SQL Server Database Wizard to quickly create a SQL Server database and an Access project at the same time. Working with an Access project is similar to working with a [Microsoft Access database](#)— the process of creating forms, reports, data access pages, macros, and modules is the same. Once you connect to a SQL Server database, you can view, create, modify, and delete tables, views, stored procedures, and database diagrams using the Microsoft SQL Server Design Tools.
- **Create a new Access database from data in another file format** You can open a file that is in another file format— such as text, dBASE, Paradox, or spreadsheet format— in Access; Microsoft Access automatically creates an Access database and links the file for you.
- **Import or link data from Microsoft Outlook or Microsoft Exchange** You can use the Exchange Import Wizard or the Outlook Import Wizard to import or link data from Microsoft Outlook and Microsoft Exchange Server. For example, you might want to link to your Microsoft Outlook Contacts folder and then create form letters and mailing labels by merging the data with the Microsoft Word Mail Merge Wizard.
- **Export to other sources** You can export data to a number of formats and applications from Access.

[Data formats you can export to](#)

Application

Version or format supported

2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access

Microsoft Access database	2002-2003
Microsoft Access project	9.0/2000, 10.0/Access 2002-2003
dBASE	III, IV, and 5; and 7 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Microsoft Excel	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2003
Microsoft Word, Rich Text Format	All
Lotus 1-2-3	.wj2, .wk1 and .wk3 formats
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML and IDC/HTX	1.0 (if a list) 2.0, 3.x, 4.x (if a table or list)
Microsoft Active Server Pages	All
XML Documents	All
SQL tables, Microsoft Visual FoxPro, and programs and databases that support the ODBC protocol	Visual FoxPro 3.0, 5.0, and 6.x For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

- **Import or link data to an Access database** Just as you can export to a number of formats and applications, you can also work with the data from other sources and applications.

 [Data formats you can import or link to](#)

Data source

Version or format supported

2.0, 7.0/95, 8.0/97, 9.0/2000,

Microsoft Access database	10.0/Access 2002-2003
Microsoft Access project	9.0/2000, 10.0/Access 2002-2003
dBASE	III, IV, and 5; and 7 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Excel spreadsheets	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2003
Lotus 1-2-3 spreadsheets (Linking is read-only)	.wks, .wk1, .wk3, and .wk4
Microsoft Exchange	All versions
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML	1.0 (if a list)
	2.0, 3.x (if a table or list)
XML Documents	All versions
SQL tables, Microsoft Visual FoxPro, and data from other programs and databases that support the ODBC protocol	Visual FoxPro 2.x, 3.0, 5.0, and 6.x (import only) For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

[Using a database for the first time](#)

Working with databases and database objects can be a daunting task when you first get started. The following information should help you become more familiar with the components that make up a Microsoft Access database.

[Databases: What they are and how they work](#)

A database is a collection of information that's related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. If your database isn't stored on a computer, or only parts of it are, you may be tracking information from a variety of sources that you're having to coordinate and organize yourself.

For example, suppose the phone numbers of your suppliers are stored in various locations: in a card file containing supplier phone numbers, in product information files in a file cabinet, and in a spreadsheet containing order information. If a supplier's phone number changes, you might have to update that information in all three places. In a database, however, you only have to update that information in one place—the supplier's phone number is automatically updated wherever you use it in the database.

Access database files

Using Microsoft Access, you can manage all your information from a single database file. Within the file, you can use:

- Tables to store your data.
- Queries to find and retrieve just the data you want.
- Forms to view, add, and update data in tables.
- Reports to analyze or print data in a specific layout.
- Data access pages to view, update, or analyze the database's data from the Internet or an intranet.



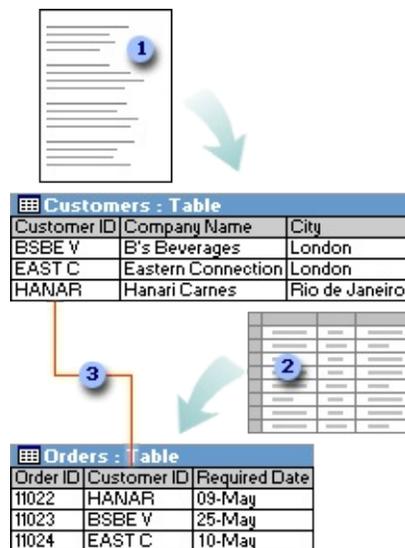
The screenshot displays several Microsoft Access objects:

- Customers : Table**: A table with columns Customer ID, Company Name, and City. Data rows include BSBE V (B's Beverages 1, London) and EAST C (Eastern Connection, London).
- London Orders for April : Select Query**: A query with columns Company Name, City, and Order. Data rows include B's Beverages 2 (London, 11-A) and Eastern Connection (London, 12-A).
- Customers : Form**: A form titled Customers with input fields for Customer ID (BSBE V) and Company Name (B's Beverages 3).
- Sales by Customer : Report**: A report titled Customer: B's Beverages 4.
- Orders by Customer : Data Access**: A data access page with a search form for Customer (Around the Horn) and another for Customer (B's Beverages 5). Below the form is a table with columns Order ID and Total, showing a row with Order ID 10289 and Total \$479.00.

- 1 Store data once in one table, but view it from multiple locations. When you update the data, it's automatically updated everywhere it appears.
- 2 Display data in a query
- 3 Display data in a form
- 4 Display data in a report
- 5 Display data in a data access page

Tables and relationships

To store your data, create one table for each type of information that you track. To bring the data from multiple tables together in a query, form, report, or data access page, define relationships between the tables.



- 1 Customer information that once existed in a mailing list now resides in the Customers table.
- 2 Order information that once existed in a spreadsheet now resides in the Orders table.
- 3 A unique ID, such as a Customer ID, distinguishes one record from another within a table. By adding one table's unique ID field to another table and defining a relationship, Microsoft Access can match related records from both

tables so that you can bring them together in a form, report, or query.

Queries

To find and retrieve just the data that meets conditions that you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform predefined or custom calculations on your data.

Customer ID	Company Name	City
BSBE V	B's Beverages	London
EAST C	Eastern Connection	London
HANAR	Hanari Carnes	Rio de Janeiro

Order ID	Required Date	Customer ID
10931	21-Apr	HANAR
10943	05-Apr	BSBE V
10987	25-Apr	EAST C

Order ID	Required Date	Company Name	City
10943	05-Apr	B's Beverages	London
10987	25-Apr	Eastern Connection	London

1 This query accesses separate tables to retrieve the Order ID, Required Date, Company Name, and City information for customers in London whose orders were required in April.

Forms

To easily view, enter, and change data directly in a table, create a form. When you open a form, Microsoft Access retrieves the data from one or more tables, and displays it on the screen with the layout you choose in the Form Wizard, or with the layout that you created on your own in [Design view](#).

Order ID	Customer ID	Required Date	Employee
11022	HANAR	09-May	Dodsworth, Anne
11023	BSBE V	25-May	Davolio, Nancy

INVOICE

Bill To: B's Beverages Fanulteroy Circus London EC25NT UK	Print Invoice												
Salesperson: Davolio, Nancy													
Order ID: 11023	Required Date: 10-May												
<table border="1"><thead><tr><th>Product</th><th>Unit Price</th><th>Quantity</th><th>Extended Price</th></tr></thead><tbody><tr><td>Chai</td><td>\$18.00</td><td>10</td><td>\$180.00</td></tr><tr><td>Ipoh Coffee</td><td>\$46.00</td><td>30</td><td>\$1380.00</td></tr></tbody></table>	Product	Unit Price	Quantity	Extended Price	Chai	\$18.00	10	\$180.00	Ipoh Coffee	\$46.00	30	\$1380.00	
Product	Unit Price	Quantity	Extended Price										
Chai	\$18.00	10	\$180.00										
Ipoh Coffee	\$46.00	30	\$1380.00										

- 1 A table displays many records at the same time, but you might have to scroll to see all of the data in a single record. Also, when viewing a table, you can't update data from more than one table at the same time.
- 2 A form focuses on one record at a time, and it can display fields from more than one table. It can also display pictures and other objects.
- 3 A form can contain a button that prints, opens other objects, or otherwise automates tasks.

[Reports](#)

To analyze your data or present it a certain way in print, create a report. For example, you might print one report that groups data and calculates totals, and another report with different data formatted for printing mailing labels.



- 1 Use a report to create mailing labels.
- 2 Use a report to show totals in a chart.
- 3 Use a report to calculate totals.

[Data access pages](#)

To make data available on the Internet or an intranet for interactive reporting, data entry, or data analysis, use a data access page. Microsoft Access retrieves the data from one or more tables and displays it on the screen with the layout

that you created on your own in Design view, or with the layout you chose in the Page Wizard.



- 1 Click the expand indicator ...
- 2 ... to display the data and record navigation toolbar for the next level of detail.
- 3 Use the record navigation toolbars to move to, sort, and filter records, and to get Help.

[Tables: Store and manage your data](#)

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means that you store that data only once. This results in a more efficient database and fewer data-entry errors.

[How data is organized in tables](#)

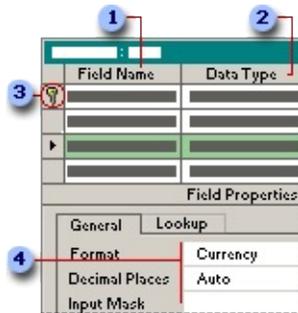
Tables organize data into columns (called fields) and rows (called records).

Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camaron Tiger	2	53

For example, each field in a Products table contains the same type of information for every product, such as the product's name. Each record in that table contains all the information about one product, such as the product's name, supplier ID number, units in stock, and so on.

Table Design view

In table Design view, you can create an entire table from scratch, or add, delete, or customize the fields in an existing table.



- 1 If you want to track additional data in a table, add more fields. If an existing field name isn't descriptive enough, you can rename the field.
- 2 Setting a field's [data type](#) defines what kind of values you can enter in a field. For example, if you want a field to store numerical values that you can use in calculations, set its data type to **Number** or **Currency**.
- 3 You use a unique tag, called a [primary key](#), to identify each record in your table. A table's primary key is used to refer to related records in other tables.
- 4 Field properties are a set of characteristics that provide additional control over how the data in a field is stored, entered, or displayed. Which properties are available depends on a field's data type.

How to relate two tables

A common field relates two tables so that Microsoft Access can bring together the data from the two tables for viewing, editing, or printing. In one table, the field is a [primary key](#) that you set in table [Design view](#). That same field also exists in the related table as a [foreign key](#).

Suppliers : Table		
Supplier ID	Company Name	
1	Exotic Liquid	
2	New Orleans	
3	Grandma Kell	
4	Tokyo Trade	

Products : Table		
Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camarvon Tiger	2	53

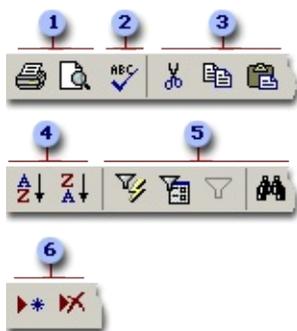
- 1 In the Suppliers table, you enter a supplier ID, company name, and so on, for each supplier. SupplierID is the primary key that you set in table Design view.
- 2 In the Products table, you include the SupplierID field, so that when you enter a new product, you can identify its supplier by entering that supplier's unique ID number. SupplierID is the foreign key in the Products table.

[Table Datasheet view](#)

In a table or query, [Datasheet view](#) provides the tools you need to work with data.

[Using the Table Datasheet and Query Datasheet toolbars](#)

The **Table Datasheet** and **Query Datasheet toolbars** provide many of the tools you need to find, edit, and print records.



- 1 Print or preview data
- 2 Check spelling
- 3 Cut, copy, or paste selected text, fields, whole records, or the entire datasheet

- 4 Sort records
- 5 Filter records, and find or replace values
- 6 Add or delete records

[Working with columns, rows, and subdatasheets](#)

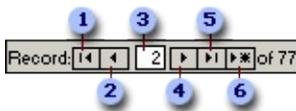
You can find tools for working with columns, rows, and [subdatasheets](#) in the datasheet itself, or by right-clicking a [column selector](#).



- 1 Use the column selector to move, hide, or rename a column.
- 2 Resize columns or rows.
- 3 Use subdatasheets to view related data.
- 4 Freeze the leftmost column so that it is displayed as you scroll to the right.

[Moving through records](#)

You can use the navigation toolbar to move through the records in a datasheet.



- 1 Go to the first record.
- 2 Go to the previous record.
- 3 Type a record number to move to.
- 4 Go to the next record.

- 5 Go to the last record.
- 6 Go to a blank (new) record.

[Queries: Getting answers to your questions](#)

You use queries to view, change, and analyze data in different ways. You can also use them as a source of records for forms, reports, and [data access pages](#). There are several types of queries in Microsoft Access.

[Select queries](#)

A select query is the most common type of query. It retrieves data from one or more tables and displays the results in a datasheet where you can update the records (with some restrictions). You can also use a select query to group records and calculate sums, counts, averages, and other types of totals.

[Parameter queries](#)

A parameter query is a query that when run displays its own dialog box prompting you for information, such as [criteria](#) for retrieving records or a value you want to insert in a field. You can design the query to prompt you for more than one piece of information; for example, you can design it to prompt you for two dates. Access can then retrieve all records that fall between those two dates.

Parameter queries are also handy when used as the basis for forms, reports, and data access pages. For example, you can create a monthly earnings report based on a parameter query. When you print the report, Access displays a dialog box asking for the month that you want the report to cover. You enter a month and Access prints the appropriate report.

[Crosstab queries](#)

You use crosstab queries to calculate and restructure data for easier analysis of your data. Crosstab queries calculate a sum, average, count, or other type of total for data that is grouped by two types of information— one down the left side of the datasheet and another across the top.

Action queries

An action query is a query that makes changes to or moves many records in just one operation. There are four types of action queries:

- **Delete Queries** A delete query deletes a group of records from one or more tables. For example, you could use a delete query to remove products that are discontinued or for which there are no orders. With delete queries, you always delete entire records, not just selected fields within records.
- **Update Queries** An update query makes global changes to a group of records in one or more tables. For example, you can raise prices by 10 percent for all dairy products, or you can raise salaries by 5 percent for the people within a certain job category. With an update query, you can change data in existing tables.
- **Append Queries** An append query adds a group of records from one or more tables to the end of one or more tables. For example, suppose that you acquire some new customers and a database containing a table of information on those customers. To avoid typing all this information into your own database, you'd like to append it to your Customers table.
- **Make-Table Queries** A make-table query creates a new table from all or part of the data in one or more tables. Make-table queries are helpful for creating a table to export to other [Microsoft Access databases](#) or a history table that contains old records.

SQL queries

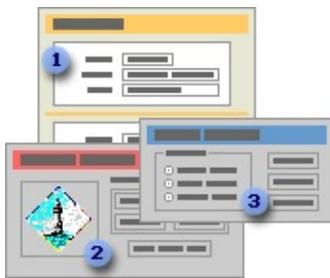
An SQL query is a query you create by using an [SQL statement](#). You can use Structured Query Language (SQL) to query, update, and manage relational databases such as Access.

When you create a query in query [Design view](#), Access constructs the equivalent SQL statements behind the scenes for you. In fact, most query properties in the property sheet in query Design view have equivalent clauses and options available in [SQL view](#). If you want, you can view or edit the SQL statement in SQL view. However, after you make changes to a query in SQL view, the query might not be displayed the way it was previously in Design view.

Some SQL queries, called [SQL-specific queries](#), can't be created in the [design grid](#). For [pass-through](#), [data-definition](#), and [union queries](#), you must create the SQL statements directly in SQL view. For [subqueries](#), you enter the SQL in the **Field** row or the **Criteria** row of the query design grid.

[Forms: Putting a friendly face on your data](#)

A form is a type of a database object that is primarily used to enter or display data in a database. You can also use a form as a switchboard that opens other forms and reports in the database, or as a custom dialog box that accepts user input and carries out an action based on the input.



- 1 A data-entry form
- 2 A switchboard form
- 3 A custom dialog box

Most forms are bound to one or more tables and [queries](#) in the database. A form's [record source](#) refers to the fields in the underlying tables and queries. A form need not contain all the fields from each of the tables or queries that it is based on.

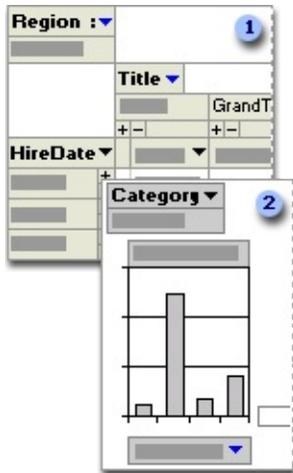
A bound form stores or retrieves data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the form's design.

- 1 Graphic elements, such as lines and rectangles, are stored in the form's design.
- 2 Data comes from the fields in the underlying record source.
- 3 A calculation comes from an expression, which is stored in the form's design.
- 4 Descriptive text is stored in the form's design.

You create a link between a form and its record source by using graphical objects called [controls](#). The most common type of control used to display and enter data is a text box.

- 1 Labels display descriptive text.
- 2 Text boxes display data from and enter data into the Products table.
- 3 A text box uses an expression to calculate a total.

You can also open a form in PivotTable view or PivotChart view to analyze data. In these views, you can dynamically change the layout of a form to present data in different ways. You can rearrange row headings, column headings, and filter fields until you achieve the desired layout. Each time you change the layout, the form immediately recalculates the data based on the new arrangement.



- 1 In PivotTable view, you can view detail or summarized data by arranging fields in the filter, row, column, and detail areas.
- 2 In PivotChart view, you can display data visually by selecting a chart type and viewing data by arranging fields in the filter, series, category, and data areas.

[Creating a form](#)

You can create a form quickly by using the AutoForm command or a wizard. AutoForm creates a form that displays all fields and records in the underlying table or [query](#). A wizard asks you questions and creates a form based on your answers. You can then customize the form the way you want it in [Design view](#).

[Customizing a form](#)

[In Design view](#)

You can customize a form in Design view in the following ways:

Record source Change the tables and queries that a form is based on.

Controlling and assisting the user You can set form properties to allow or prevent users from adding, deleting, or editing records displayed in a form. You can also add custom Help to a form to assist your users with using the form.

Form window You can add or remove **Maximize** and **Minimize** buttons, short cut menus, and other Form window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a form. You can also set section properties to control the appearance and printing of a form.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a form.

 [In PivotTable or PivotChart view](#)

You can customize a form in PivotTable or PivotChart view in the following ways:

Add, move, or remove fields You can add fields to the filter, row, column, and detail areas in PivotTable view, and to the filter, category, series, and data areas in PivotChart view. You can also move fields from one area to another and remove fields from the view.

Filter records You can filter data displayed in the view by adding or moving a field to the filter area. You can also filter a field in the row and column area.

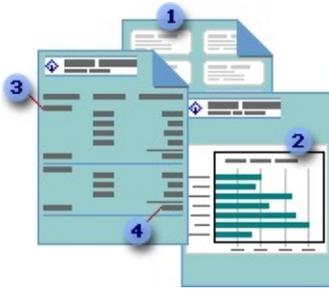
Sort records You can sort items in row or column fields in ascending or descending order. You can also sort items in custom order in PivotTable view.

Group records You can group items in row or column fields on intervals, or create custom groups.

Format elements and change captions In PivotTable view, you can change the font settings, alignment, background color, and number format of a field. You can also change the captions of fields and custom groups. In PivotChart view, you can change the chart type, format data markers, and more.

 [Reports: Presenting your data to the world](#)

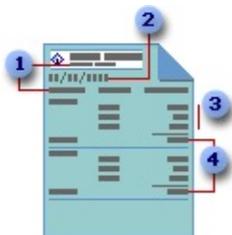
A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.



- 1 Create mailing labels.
- 2 Show totals in a chart.
- 3 Group records into categories.
- 4 Calculate totals.

Most reports are bound to one or more table and [query](#) in the database. A report's record source refers to the fields in the underlying tables and queries. A report need not contain all the fields from each of the tables or queries that it is based on.

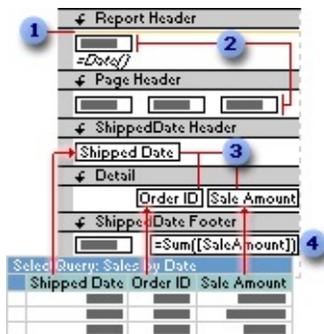
A bound report gets its data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the report's design.



- 1 The report title and column headings are stored in the report's design.
- 2 The date comes from an expression stored in the report's design.
- 3 Data comes from fields in the underlying table, query, or SQL statement.
- 4 Totals come from expressions, which are stored in the report's design.

You create the link between a report and its record source by using graphical

objects called [controls](#). Controls can be text boxes that display names and numbers, labels that display titles, or decorative lines that graphically organize the data and make the report more attractive.



- 1 Lines are decorative.
- 2 Labels display descriptive text.
- 3 Text boxes display data from the Sales by Date query.
- 4 A text box uses an expression to calculate a total.

[Creating a report](#)

You can create different types of reports quickly by using wizards. Use the Label Wizard to create mailing labels, the Chart Wizard to create charts, or the Report Wizard to create a standard report. The wizard asks you questions and creates a report based on your answers. You can then customize the report the way you want it in [Design view](#).

[Customizing a report](#)

You can customize a report in the following ways:

Record source Change the tables and queries that a report is based on.

Sorting and grouping data You can sort data in ascending or descending order. You can also group records on one or more fields, and display subtotals and grand totals on a report.

Report window You can add or remove **Maximize** and **Minimize** buttons,

change the title bar text, and other Report window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a report. You can also set section properties to control the appearance and printing of a report.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a report.

[Data access pages: Publishing your data to the Web](#)

A data access page is a special type of Web page designed for viewing and working with data from the Internet or an intranet— data that is stored in a [Microsoft Access database](#) or a [Microsoft SQL Server database](#). The data access page may also include data from other sources, such as Microsoft Excel.

[A page in Page view or Microsoft Internet Explorer 5.01 with Service Pack 2 \(SP2\) or later](#)

Using a data access page is similar to using a form: You can view, enter, edit, and delete data in a database. However, you can also use a page outside a [Microsoft Access database](#), so users can update or view data over the Internet or an intranet.

The following illustration shows how the Product data access page, which is used for entering and updating product information, looks in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.



- 1 Enter product information in the text boxes, drop-down list box, and check box.
- 2 Use the record navigation toolbar to navigate, add, delete, save, sort, and filter records, and to get Help.

 [A page in Design view](#)

The following illustration shows the Products data access page in [Design view](#).



- 1 Title in the body of the page
- 2 Group header for the Products group level
- 3 Record navigation section for the Products group level

 [Parts of a data access page](#)

The body The body is the basic design surface of a data access page. On a page that supports data entry, you can use it to display informational text, controls bound to data, and sections.

Sections You use sections to display text, data from a database, and toolbars.

Two types of [sections](#) are typically used on pages that support data entry: group header and record navigation sections. A page can also have footer and caption sections.

- **Group header and footer** Used to display data and calculate values.
- **Record navigation** Used to display the [record navigation control](#) for the group level. A record navigation section for a group appears after the group header section. You can't place [bound controls](#) in a record navigation section.
- **Caption** Used to display captions for text boxes and other controls. It appears immediately before the group header. You can't place bound controls in a caption section.

Each group level in a data access page has a [record source](#). The name of the record source is displayed on the [section bar](#) for each section used for a group level.

[Designing different types of data access pages](#)

You design data access pages in [Design view](#) in Microsoft Access. The page is a separate file that is stored outside Access; however, when you create the file, Access automatically adds a shortcut to the file in the Database window.

Designing a data access page is similar to designing forms and reports—you use a [field list](#), the [toolbox](#), [controls](#), and so on. However, there are some significant differences in the way that you design and interact with data access pages as opposed to forms and reports. How you design the page depends on what it will be used for.

- **Interactive reporting** This type of data access page is often used to consolidate and group information that is stored in the database, and then publish summaries of the data. For example, a page might publish the sales performance for each region in which you do business. Using [expand indicators](#), you can go from a general summary of the information, such as a list of all the regions and their combined sales total, to specific details on individual sales within each region. The data access page might provide toolbar buttons for sorting and [filtering](#) the data, as well as for adding, editing, and deleting the data in some or all [group levels](#).
- **Data analysis** This type of data access page may include a [PivotTable list](#), similar to a Microsoft Excel [PivotTable report](#), that lets you reorganize the data to analyze it in different ways. The page might contain a chart that you can use to analyze trends, detect patterns, and compare data in your database. Or it might contain a spreadsheet, in which you can enter and edit data and use formulas to calculate as you do in Excel.

[Using data access pages in Internet Explorer](#)

A data access page is connected directly to a database. When users display the data access page in Internet Explorer, they are viewing their own copy of the page. That means any filtering, sorting, and other changes they make to the way the data is displayed—including changes they make within a [PivotTable list](#) or [spreadsheet](#)—affect only their copy of the data access page. However, changes that they make to the data itself—such as modifying values, and adding or deleting data—are stored in the underlying database, and therefore are available to everyone viewing the data access page.

Users get Help on how to work with the page in Internet Explorer by clicking the **Help**  button on the record navigation toolbar. The Help file that appears is automatically included with any data access page that's published with a record navigation toolbar. If you delete the record navigation toolbar or if you disable its **Help** button, you should provide instructions for using the page to those who will use the page.

Note To view and work with the data access page on the Internet or an intranet, users need Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

[Using data access pages in Microsoft Access](#)

You can also work with a data access page in [Page view](#) in Access. Data access pages can supplement the forms and reports that you use in your database application. When deciding whether to design a data access page, form, or report, consider the tasks that you want to perform.

You get Help on how to work with the page in Page view by clicking the **Help**  button on the **Page View** toolbar to display Access Help. Clicking the **Help** button on a record navigation toolbar on the page displays Help for working with the page in Microsoft Internet Explorer. As mentioned previously, this Help file is automatically included with any data access page that's published with a record navigation toolbar. You can delete the **Help** button on pages that are used only within Access, or you can modify the button to provide custom Help.

[Where a data access page gets its data](#)

[Types of data sources for a data access page](#)

Microsoft Access database, or a [Microsoft SQL Server database](#) version 6.5 or later. To design a page using data from one of these databases, the page must be connected to the database. If you have already opened an Access database or an [Access project](#) that's connected to a SQL Server database, the data access page you create automatically connects to the current database and stores that path in the **ConnectionString** property of the data access page. When a user browses to the page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later or displays the page in [Page view](#), it displays current data from the underlying database by using the path defined in the **ConnectionString** property. If you design the data access page while the database is on a local drive, Access will use the local path, which means the data won't be accessible to others. For this reason, it's important that you move or copy the database to a network location that's accessible to users of your page. Once the database is on the network share, open the database using the [UNC](#) address. If you move or copy the database after you've already designed the page, you'll have to update the path in the **ConnectionString** property to point to the new location.

Instead of having to update the **ConnectionString** property of each page in the database separately, you can choose to create a connection file. A connection file stores the connection information for a data access page, and can be shared between multiple data access pages. When you open a data access page that uses a connection file, the page reads the connection file and connects to the appropriate database. After creating the connection file, if you move or copy the database, all you have to do is edit the connection information in the connection file.

[Types of data sources for Microsoft Office Web Components on a data access page](#)

Although the data access page gets its data from a [Microsoft Access database](#) or [Microsoft SQL Server database](#), [Microsoft Office Web Component](#) controls on the page can display data from either of these databases or from other sources. For example, the page may contain a [PivotTable list](#), [spreadsheet](#), or [chart](#) with data from a Microsoft Excel worksheet or a database other than Access or SQL Server. Depending on the data source, the data in these controls may be a snapshot of the original data or it may be live data displayed through a separate

connection that the control has with its data source.

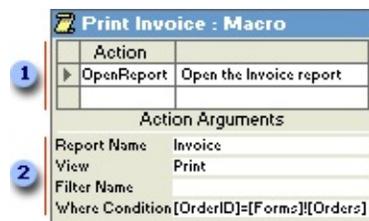
To create an Office Web Component with data from a source other than the current database, you might have to use a program other than Access to first publish the data to a Web page. For example, to create a PivotTable list that contains data from an Excel worksheet, you have to publish the data from Excel. Then, in Access, you can open the Web page and turn it into a data access page by adding controls that are bound to an Access or SQL Server database. You can further customize the page by adding labels, pictures, or other features to enhance the page's appearance. From within Access, you can create a PivotTable list that uses data from an external source other than Excel.

[Macros: Putting your data to work](#)

Macros are a set of actions that you can create to help you to automate common tasks. By using groups of macros, you can perform several tasks at once.

[What is a macro?](#)

A macro is a set of one or more [actions](#) that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.



1 When you create a macro, you enter the actions you want to carry out in this portion of the [Macro window](#).

2 You can specify arguments for an action in this portion of the window.

A macro can be one macro composed of a sequence of actions, or it can be a [macro group](#). You can also use a [conditional expression](#) to determine whether in some cases an action will be carried out when a macro runs.

The following macro is composed of a series of actions. Microsoft Access carries out these actions each time the macro runs. To run this macro, you refer to the macro name Review Products.



Action	Comment
Echo	Attached to the Add
OpenForm	Freeze screen while
MoveSize	Open the Product List Position the Product L

[What is a macro group?](#)

If you have numerous macros, grouping related macros in [macro groups](#) can help you to manage your database more easily.

For example, the following macro group, named Buttons, is made up of three related macros: Employees, Products, and Reps. Each macro carries out the OpenForm action, and the Products macro also carries out the MoveSize action.



Macro Name	Action
Employees	OpenForm
Products	OpenForm
	MoveSize
Reps	OpenForm

The name in the **Macro Name** column identifies each macro. When you run a macro in a macro group, Microsoft Access carries out the action in the action column and any actions that immediately follow with a blank **Macro Name** column.

You can run a macro in a macro group in an event or [event procedure](#) by typing the macro group name followed by a period and then the macro name. In the preceding example, to refer to the Employees macro in the Buttons macro group, you would type **Buttons.Employees**.

[Conditional actions](#)

In some cases, you may want to carry out an action or series of actions in a macro only if a particular [condition](#) is true. For example, if you're using a macro to validate data in a form, you might want to display one message in response to one set of values entered in a record and another message in response to a

different set of values. In cases like these, you can use conditions to control the flow of the macro.

A condition is a logical [expression](#) that evaluates to True/False or Yes/No. The macro follows different paths depending on whether the condition is true or false.

When you run the macro, Microsoft Access evaluates the first [conditional expression](#). If the condition is true, Microsoft Access carries out the action in that row and any of the immediately following actions that are preceded by an ellipsis (...) in the **Condition** column.

Microsoft Access then runs any additional actions in the macro that have a blank **Condition** column until it reaches another expression, a macro name, or the end of the macro.

If the condition is false, Microsoft Access ignores the action and any immediately following actions that are preceded by an ellipsis in the **Condition** column and moves to the next [action row](#) that contains another condition or a blank **Condition** column.

The following macro runs the MsgBox and the StopMacro actions only when the expression in the **Condition** column is true (when there is a [Null](#) value in the SupplierID field).



Condition	Action
IsNull([SupplierID])	MsgBox
	StopMacro

[Action arguments](#)

Action arguments are additional information required by some macro actions — for example, the object affected by the action or special conditions under which the action is carried out. After you add an action to a macro, you set the arguments for the action in the lower portion of the [Macro window](#). These arguments give Microsoft Access additional information on how to carry out the action.

Tips for setting action arguments:

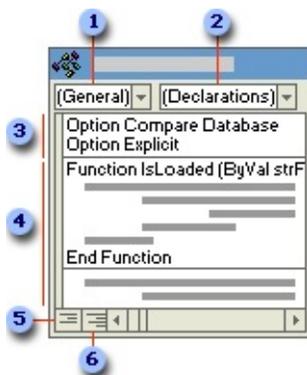
- In general, it's a good idea to set action arguments in the order they're listed, because choices for one argument may determine those for arguments that follow.
- If you add an action to your macro by dragging a [database object](#) from the [Database window](#), Microsoft Access automatically sets appropriate arguments for that action.
- If an action has an argument that calls for the name of a database object, you can set the argument and the corresponding object type argument automatically by dragging the object from the Database window to the argument box.
- You can use an [expression](#) preceded by an equal sign (=) to set many action arguments.

[Modules: Automate routine tasks and create business solutions](#)

A module is essentially a collection of declarations, statements, and procedures stored together as one named unit to organize your [Microsoft Visual Basic](#) code. Microsoft Access has two types of modules: [standard modules](#) and [class modules](#).

[What is a module?](#)

A module is a collection of Visual Basic declarations and procedures that are stored together as a unit.



1 Object box

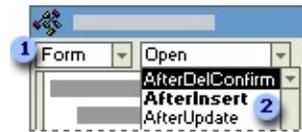
2 Procedure box

- 3 Declarations
- 4 Procedures
- 5 **Procedure View** button
- 6 **Full Module View** button

[Class modules](#)

[Form](#) and [report modules](#) are class modules that are associated with a particular form or report. Form and report modules often contain [event procedures](#) that run in response to an event on the form or report. You can use event procedures to control the behavior of your forms and reports, and their response to user actions, such as clicking a command button.

When you create the first event procedure for a form or report, Microsoft Access automatically creates an associated form or report module.



- 1 To view or add a form procedure, choose one of the form's objects from the **Object** box ...
- 2 ... and then choose an event from the **Procedure** box. Names of events that already have procedures are bold.

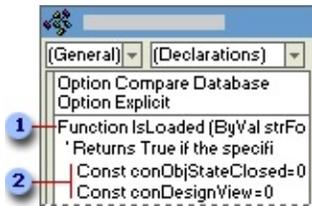
Procedures in your form and report modules can call procedures you have added to standard modules.

In Access 97 or later, class modules can also exist independent of a form or report, and this type of class module is listed in the [Database window](#). You can use a class module to create a definition for a custom object. In Access 95, class modules exist in association only with a form or report.

[Standard modules](#)

Standard modules contain general procedures that aren't associated with any

other object, and frequently used procedures that can be run from anywhere within your database. The major difference between a standard module and a class module that is not associated with a particular object is one of scope and lifetime. The value of any variables or constants that are declared or exist in a class module without an associated object are available for use only while that code is running and only from that object.



- 1 Function name and arguments
- 2 Visual Basic declarations and code statements

Standard modules are listed in **Modules** under **Objects** in the [Database window](#). Form, report, and standard modules are also listed in the [Object Browser](#).

[Relationships in a database](#)

After you've set up different tables for each subject in your [Microsoft Access database](#), you need a way of telling Microsoft Access how to bring that information back together again. The first step in this process is to define [relationships](#) between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once. For example, this form includes information from four tables:



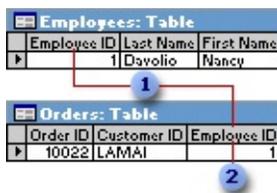
- 1 The Customers table
- 2 The Orders table

3 The Products table

4 The Order Details table

How relationships work

In the previous example, the fields in four tables must be coordinated so that they show information about the same order. This coordination is accomplished with relationships between tables. A relationship works by matching data in key fields— usually a field with the same name in both tables. In most cases, these matching fields are the [primary key](#) from one table, which provides a unique identifier for each record, and a [foreign key](#) in the other table. For example, employees can be associated with orders they're responsible for by creating a relationship between the EmployeeID fields.



1 EmployeeID appears in both tables— -as a primary key ...

2 ... and as a foreign key.

A one-to-many relationship

A one-to-many relationship is the most common type of relationship. In a one-to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.

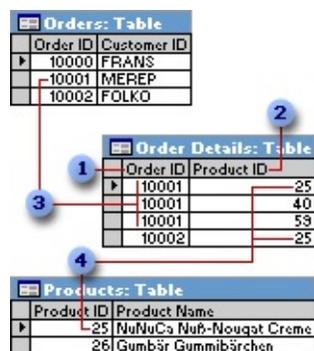


1 One supplier ...

- 2 ... can supply more than one product ...
- 3 ... but each product has only one supplier.

[A many-to-many relationship](#)

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose [primary key](#) consists of two fields— the [foreign keys](#) from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table. For example, the Orders table and the Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.



- 1 Primary key from the Orders table
- 2 Primary key from the Products table
- 3 One order can have many products ...
- 4 ... and each product can appear on many orders.

[A one-to-one relationship](#)

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one

relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. For example, you might want to create a table to track employees participating in a fundraising soccer game. Each soccer player in the Soccer Players table has one matching record in the Employees table.

Employee ID	Last Name	First Name
1	Davolio	Nancy
2	Fuller	Andrew
3	Leverling	Janet
4	Peacock	Margaret
5	Buchanan	Steven

Employee ID	Player Nickname	Skill Level
1	Slammin' Man	2
3	Ace	1
5	Stevemeister	2

- 1 Each soccer player has one matching record in the Employees table.
- 2 This set of values is a subset of the EmployeeID field and the Employees table.

[About defining relationships](#)

The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

- A one-to-many relationship is created if only one of the related fields is a [primary key](#) or has a [unique index](#).
- A one-to-one relationship is created if both of the related fields are primary keys or have unique indexes.
- A many-to-many relationship is really two one-to-many relationships with a third table whose primary key consists of two fields—the [foreign keys](#) from the two other tables.

You can also create a relationship between a table and itself. This is useful in situations where you need to perform a Lookup within the same table. In the Employees table, for example, you can define a relationship between the EmployeeID and ReportsTo fields, so that the ReportsTo field can display employee data from a matching EmployeeID.

Note If you drag a field that isn't a primary key and doesn't have a unique index to another field that isn't a primary key and doesn't have a unique index, an indeterminate relationship is created. In queries containing tables with an indeterminate relationship, Microsoft Access displays a default [join](#) line between

the tables, but [referential integrity](#) won't be enforced, and there's no guarantee that records are unique in either table.

[Referential integrity](#)

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. You can set referential integrity when all of the following conditions are met:

- The matching field from the [primary table](#) is a [primary key](#) or has a [unique index](#).
- The related fields have the same [data type](#). There are two exceptions. An [AutoNumber](#) field can be related to a Number field with a **FieldSize** property setting of **Long Integer**, and an AutoNumber field with a **FieldSize** property setting of **Replication ID** can be related to a Number field with a **FieldSize** property setting of **Replication ID**.
- Both tables belong to the same Microsoft Access database. If the tables are linked tables, they must be tables in Microsoft Access format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

The following rules apply when you use referential integrity:

- You can't enter a value in the [foreign key](#) field of the related table that doesn't exist in the primary key of the primary table. However, you can enter a **Null** value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.
- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

[Cascading updates and deletes](#)

For relationships in which [referential integrity](#) is enforced, you can specify whether you want Microsoft Access to automatically [cascade update](#) and [cascade delete](#) related records. If you set these options, delete and update operations that would normally be prevented by referential integrity rules are allowed. When you delete records or change [primary key](#) values in a [primary table](#), Microsoft Access makes necessary changes to related tables to preserve referential integrity.

If you select the **Cascade Update Related Fields** check box when defining a relationship, any time you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship isn't broken. Microsoft Access cascades updates without displaying any message.

Note If the primary key in the primary table is an [AutoNumber](#) field, setting the **Cascade Update Related Fields** check box will have no effect, because you can't change the value in an AutoNumber field.

If you select the **Cascade Delete Related Records** check box when defining a relationship, any time you delete records in the primary table, Microsoft Access automatically deletes related records in the related table. For example, if you delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the **Cascade Delete Related Records** check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a [delete query](#), Microsoft Access automatically deletes the records in related tables without displaying a warning.

[Show All](#)

Ways to get started if you've used other database or spreadsheet applications

If you are familiar with other database or spreadsheet applications, then you probably know the basics of how these applications work and what databases are used for. Microsoft Access differs from many other database applications by allowing you to create [relational databases](#). Access also provides many options for working with other database programs such as Microsoft SQL Server.

[Relationships in a database](#)

After you've set up different tables for each subject in your [Microsoft Access database](#), you need a way of telling Microsoft Access how to bring that information back together again. The first step in this process is to define [relationships](#) between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once. For example, this form includes information from four tables:



Product:	Unit Price:	Quantity:
▾ Gudgrandsbrosst ▾	\$24.00	14
Scottish Longbread	\$12.50	20
Pâté chinois	\$36.00	10
*		

- 1 The Customers table
- 2 The Orders table
- 3 The Products table
- 4 The Order Details table

How relationships work

In the previous example, the fields in four tables must be coordinated so that they show information about the same order. This coordination is accomplished with relationships between tables. A relationship works by matching data in key fields— usually a field with the same name in both tables. In most cases, these matching fields are the [primary key](#) from one table, which provides a unique identifier for each record, and a [foreign key](#) in the other table. For example, employees can be associated with orders they're responsible for by creating a relationship between the EmployeeID fields.

Employees: Table		
Employee ID	Last Name	First Name
1	Davolio	Nancy

Orders: Table		
Order ID	Customer ID	Employee ID
10022	LAMAI	1

- 1 EmployeeID appears in both tables— -as a primary key ...
- 2 ... and as a foreign key.

A one-to-many relationship

A one-to-many relationship is the most common type of relationship. In a one-to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.

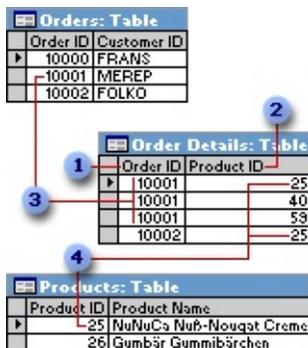
Suppliers: Table	
Supplier ID	Company Name
1	Exotic Liquids
2	New Orleans Cajun Delights
3	Grandma Kelly's Homestead
4	Tokyo Traders

Products: Table		
Product ID	Product Name	Supplier ID
1	Chai	1
2	Chang	1
3	Aniseed Syrup	1
4	Chef Anton's Cajun Seasoning	2
5	Chef Anton's Gumbo Mix	2

- 1 One supplier ...
- 2 ... can supply more than one product ...
- 3 ... but each product has only one supplier.

A many-to-many relationship

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose [primary key](#) consists of two fields—the [foreign keys](#) from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table. For example, the Orders table and the Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.



- 1 Primary key from the Orders table
- 2 Primary key from the Products table
- 3 One order can have many products ...
- 4 ... and each product can appear on many orders.

A one-to-one relationship

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. For example, you might want to create a table to track employees participating in a fundraising soccer game. Each soccer player in the Soccer

Players table has one matching record in the Employees table.

Employees: Table		
Employee ID	Last Name	First Name
1	Davolio	Nancy
2	Fuller	Andrew
3	Leverling	Janet
4	Peacock	Margaret
5	Buchanan	Steven

Soccer Players: Table		
Employee ID	Player Nickname	Skill Level
1	Slammin' Man	2
3	Ace	1
5	Stevemeister	2

- 1 Each soccer player has one matching record in the Employees table.
- 2 This set of values is a subset of the EmployeeID field and the Employees table.

[About defining relationships](#)

The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

- A one-to-many relationship is created if only one of the related fields is a [primary key](#) or has a [unique index](#).
- A one-to-one relationship is created if both of the related fields are primary keys or have unique indexes.
- A many-to-many relationship is really two one-to-many relationships with a third table whose primary key consists of two fields—the [foreign keys](#) from the two other tables.

You can also create a relationship between a table and itself. This is useful in situations where you need to perform a Lookup within the same table. In the Employees table, for example, you can define a relationship between the EmployeeID and ReportsTo fields, so that the ReportsTo field can display employee data from a matching EmployeeID.

Note If you drag a field that isn't a primary key and doesn't have a unique index to another field that isn't a primary key and doesn't have a unique index, an indeterminate relationship is created. In queries containing tables with an indeterminate relationship, Microsoft Access displays a default [join](#) line between the tables, but [referential integrity](#) won't be enforced, and there's no guarantee that records are unique in either table.

[Referential integrity](#)

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. You can set referential integrity when all of the following conditions are met:

- The matching field from the [primary table](#) is a [primary key](#) or has a [unique index](#).
- The related fields have the same [data type](#). There are two exceptions. An [AutoNumber](#) field can be related to a Number field with a **FieldSize** property setting of **Long Integer**, and an AutoNumber field with a **FieldSize** property setting of **Replication ID** can be related to a Number field with a **FieldSize** property setting of **Replication ID**.
- Both tables belong to the same Microsoft Access database. If the tables are linked tables, they must be tables in Microsoft Access format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

The following rules apply when you use referential integrity:

- You can't enter a value in the [foreign key](#) field of the related table that doesn't exist in the primary key of the primary table. However, you can enter a **Null** value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.
- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

[Cascading updates and deletes](#)

For relationships in which [referential integrity](#) is enforced, you can specify whether you want Microsoft Access to automatically [cascade update](#) and [cascade delete](#) related records. If you set these options, delete and update operations that would normally be prevented by referential integrity rules are allowed. When you delete records or change [primary key](#) values in a [primary table](#), Microsoft Access makes necessary changes to related tables to preserve referential integrity.

If you select the **Cascade Update Related Fields** check box when defining a relationship, any time you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship isn't broken. Microsoft Access cascades updates without displaying any message.

Note If the primary key in the primary table is an [AutoNumber](#) field, setting the **Cascade Update Related Fields** check box will have no effect, because you can't change the value in an AutoNumber field.

If you select the **Cascade Delete Related Records** check box when defining a relationship, any time you delete records in the primary table, Microsoft Access automatically deletes related records in the related table. For example, if you delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the **Cascade Delete Related Records** check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a [delete query](#), Microsoft Access automatically deletes the records in related tables without displaying a warning.

[Working with other applications](#)

Microsoft Access 2002 or later provides features for working with other programs.

- **Work with Microsoft SQL Server** You can create a [Microsoft Access project](#) that is easy to connect to a Microsoft SQL Server database, or use the Microsoft SQL Server Database Wizard to quickly create a SQL Server

database and an Access project at the same time. Working with an Access project is similar to working with a [Microsoft Access database](#)— the process of creating forms, reports, data access pages, macros, and modules is the same. Once you connect to a SQL Server database, you can view, create, modify, and delete tables, views, stored procedures, and database diagrams using the Microsoft SQL Server Design Tools.

- **Create a new Access database from data in another file format** You can open a file that is in another file format— such as text, dBASE, Paradox, or spreadsheet format— in Access; Microsoft Access automatically creates an Access database and links the file for you.
- **Import or link data from Microsoft Outlook or Microsoft Exchange** You can use the Exchange Import Wizard or the Outlook Import Wizard to import or link data from Microsoft Outlook and Microsoft Exchange Server. For example, you might want to link to your Microsoft Outlook Contacts folder and then create form letters and mailing labels by merging the data with the Microsoft Word Mail Merge Wizard.
- **Export to other sources** You can export data to a number of formats and applications from Access.

 [Data formats you can export to](#)

Application	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002 or later
Microsoft Access project	9.0/2000, 10.0/Access 2002 or later
dBASE	III, IV, and 5; and 7 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Microsoft Excel	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2002 or later
Microsoft Word, Rich Text Format	All

Lotus 1-2-3	.wj2, .wk1 and .wk3 formats
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML and IDC/HTX	1.0 (if a list) 2.0, 3.x, 4.x (if a table or list)
Microsoft Active Server Pages	All
XML Documents	All
SQL tables, Microsoft Visual FoxPro, and programs and databases that support the ODBC protocol	Visual FoxPro 3.0, 5.0, and 6.x For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

- **Import or link data to an Access database** Just as you can export to a number of formats and applications, you can also work with the data from other sources and applications.

 [Data formats you can import or link to](#)

Data source	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002 or later
Microsoft Access project	9.0/2000, 10.0/Access 2002 or later
dBASE	III, IV, and 5; and 7 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Excel spreadsheets	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2002 or later
Lotus 1-2-3 spreadsheets (Linking is read-only)	.wks, .wk1, .wk3, and .wk4

Microsoft Exchange	All versions
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML	1.0 (if a list) 2.0, 3.x (if a table or list)
XML Documents	All versions Visual FoxPro 2.x, 3.0, 5.0, and 6.x
SQL tables, Microsoft Visual FoxPro, and data from other programs and databases that support the ODBC protocol	(import only) For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

[Show All](#)

Ways to get started if you're using a database for the first time

Working with databases and database objects can be a daunting task when you first get started. The following information should help you become more familiar with the components that make up a Microsoft Access database.



[Databases: What they are and how they work](#)

A database is a collection of information that's related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. If your database isn't stored on a computer, or only parts of it are, you may be tracking information from a variety of sources that you're having to coordinate and organize yourself.

For example, suppose the phone numbers of your suppliers are stored in various locations: in a card file containing supplier phone numbers, in product information files in a file cabinet, and in a spreadsheet containing order information. If a supplier's phone number changes, you might have to update that information in all three places. In a database, however, you only have to update that information in one place—the supplier's phone number is automatically updated wherever you use it in the database.



[Access database files](#)

Using Microsoft Access, you can manage all your information from a single database file. Within the file, you can use:

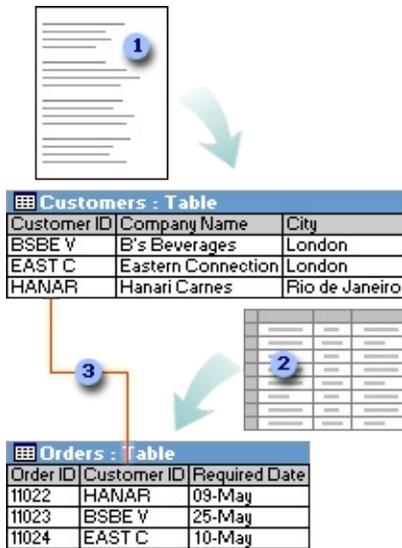
- Tables to store your data.
- Queries to find and retrieve just the data you want.
- Forms to view, add, and update data in tables.
- Reports to analyze or print data in a specific layout.
- Data access pages to view, update, or analyze the database's data from the Internet or an intranet.



- 1 Store data once in one table, but view it from multiple locations. When you update the data, it's automatically updated everywhere it appears.
- 2 Display data in a query
- 3 Display data in a form
- 4 Display data in a report
- 5 Display data in a data access page

[Tables and relationships](#)

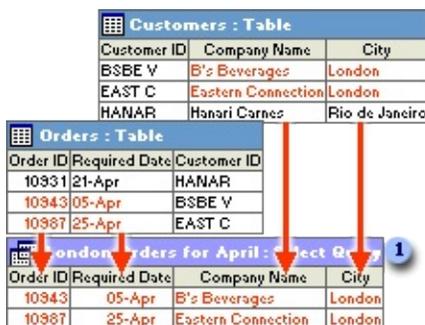
To store your data, create one table for each type of information that you track. To bring the data from multiple tables together in a query, form, report, or data access page, define relationships between the tables.



- 1 Customer information that once existed in a mailing list now resides in the Customers table.
- 2 Order information that once existed in a spreadsheet now resides in the Orders table.
- 3 A unique ID, such as a Customer ID, distinguishes one record from another within a table. By adding one table's unique ID field to another table and defining a relationship, Microsoft Access can match related records from both tables so that you can bring them together in a form, report, or query.

Queries

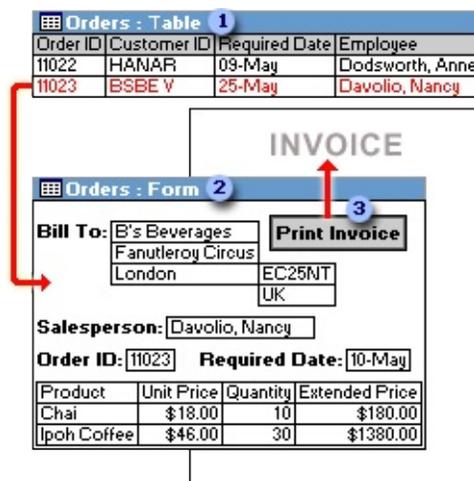
To find and retrieve just the data that meets conditions that you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform predefined or custom calculations on your data.



1 This query accesses separate tables to retrieve the Order ID, Required Date, Company Name, and City information for customers in London whose orders were required in April.

Forms

To easily view, enter, and change data directly in a table, create a form. When you open a form, Microsoft Access retrieves the data from one or more tables, and displays it on the screen with the layout you choose in the Form Wizard, or with the layout that you created on your own in [Design view](#).



1 A table displays many records at the same time, but you might have to scroll to see all of the data in a single record. Also, when viewing a table, you can't update data from more than one table at the same time.

2 A form focuses on one record at a time, and it can display fields from more than one table. It can also display pictures and other objects.

3 A form can contain a button that prints, opens other objects, or otherwise automates tasks.

Reports

To analyze your data or present it a certain way in print, create a report. For example, you might print one report that groups data and calculates totals, and another report with different data formatted for printing mailing labels.



- 1 Use a report to create mailing labels.
- 2 Use a report to show totals in a chart.
- 3 Use a report to calculate totals.

[Data access pages](#)

To make data available on the Internet or an intranet for interactive reporting, data entry, or data analysis, use a data access page. Microsoft Access retrieves the data from one or more tables and displays it on the screen with the layout that you created on your own in Design view, or with the layout you chose in the Page Wizard.



- 1 Click the expand indicator ...
- 2 ... to display the data and record navigation toolbar for the next level of detail.

3 Use the record navigation toolbars to move to, sort, and filter records, and to get Help.

[Tables: Store and manage your data](#)

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means that you store that data only once. This results in a more efficient database and fewer data-entry errors.

[How data is organized in tables](#)

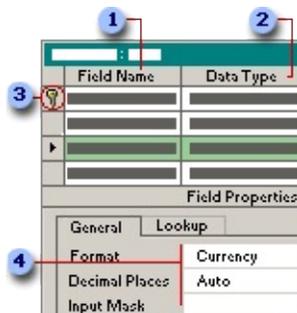
Tables organize data into columns (called fields) and rows (called records).

Products : Table		
Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camarvon Tiger	2	53

For example, each field in a Products table contains the same type of information for every product, such as the product's name. Each record in that table contains all the information about one product, such as the product's name, supplier ID number, units in stock, and so on.

[Table Design view](#)

In table Design view, you can create an entire table from scratch, or add, delete, or customize the fields in an existing table.



1 If you want to track additional data in a table, add more fields. If an existing field name isn't descriptive enough, you can rename the field.

- 2 Setting a field's [data type](#) defines what kind of values you can enter in a field. For example, if you want a field to store numerical values that you can use in calculations, set its data type to **Number** or **Currency**.
- 3 You use a unique tag, called a [primary key](#), to identify each record in your table. A table's primary key is used to refer to related records in other tables.
- 4 Field properties are a set of characteristics that provide additional control over how the data in a field is stored, entered, or displayed. Which properties are available depends on a field's data type.

[How to relate two tables](#)

A common field relates two tables so that Microsoft Access can bring together the data from the two tables for viewing, editing, or printing. In one table, the field is a [primary key](#) that you set in table [Design view](#). That same field also exists in the related table as a [foreign key](#).

Suppliers : Table	
Supplier ID	Company Name
1	Exotic Liquid
2	New Orleans
3	Grandma Kell
4	Tokyo Trade

Products : Table		
Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camarvon Tiger	2	53

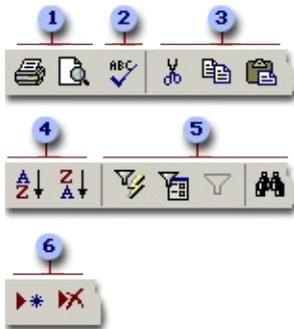
- 1 In the Suppliers table, you enter a supplier ID, company name, and so on, for each supplier. SupplierID is the primary key that you set in table Design view.
- 2 In the Products table, you include the SupplierID field, so that when you enter a new product, you can identify its supplier by entering that supplier's unique ID number. SupplierID is the foreign key in the Products table.

[Table Datasheet view](#)

In a table or query, [Datasheet view](#) provides the tools you need to work with data.

[Using the Table Datasheet and Query Datasheet toolbars](#)

The **Table Datasheet** and **Query Datasheet toolbars** provide many of the tools you need to find, edit, and print records.



- 1 Print or preview data
- 2 Check spelling
- 3 Cut, copy, or paste selected text, fields, whole records, or the entire datasheet
- 4 Sort records
- 5 Filter records, and find or replace values
- 6 Add or delete records

Working with columns, rows, and subdatasheets

You can find tools for working with columns, rows, and [subdatasheets](#) in the datasheet itself, or by right-clicking a [column selector](#).

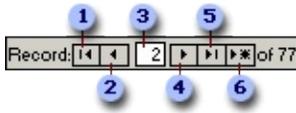


- 1 Use the column selector to move, hide, or rename a column.
- 2 Resize columns or rows.
- 3 Use subdatasheets to view related data.

- 4 Freeze the leftmost column so that it is displayed as you scroll to the right.

[Moving through records](#)

You can use the navigation toolbar to move through the records in a datasheet.



- 1 Go to the first record.
- 2 Go to the previous record.
- 3 Type a record number to move to.
- 4 Go to the next record.
- 5 Go to the last record.
- 6 Go to a blank (new) record.

[Queries: Getting answers to your questions](#)

You use queries to view, change, and analyze data in different ways. You can also use them as a source of records for forms, reports, and [data access pages](#). There are several types of queries in Microsoft Access.

[Select queries](#)

A select query is the most common type of query. It retrieves data from one or more tables and displays the results in a datasheet where you can update the records (with some restrictions). You can also use a select query to group records and calculate sums, counts, averages, and other types of totals.

[Parameter queries](#)

A parameter query is a query that when run displays its own dialog box prompting you for information, such as [criteria](#) for retrieving records or a value you want to insert in a field. You can design the query to prompt you for more

than one piece of information; for example, you can design it to prompt you for two dates. Access can then retrieve all records that fall between those two dates.

Parameter queries are also handy when used as the basis for forms, reports, and data access pages. For example, you can create a monthly earnings report based on a parameter query. When you print the report, Access displays a dialog box asking for the month that you want the report to cover. You enter a month and Access prints the appropriate report.

[Crosstab queries](#)

You use crosstab queries to calculate and restructure data for easier analysis of your data. Crosstab queries calculate a sum, average, count, or other type of total for data that is grouped by two types of information— one down the left side of the datasheet and another across the top.

[Action queries](#)

An action query is a query that makes changes to or moves many records in just one operation. There are four types of action queries:

- **Delete Queries** A delete query deletes a group of records from one or more tables. For example, you could use a delete query to remove products that are discontinued or for which there are no orders. With delete queries, you always delete entire records, not just selected fields within records.
- **Update Queries** An update query makes global changes to a group of records in one or more tables. For example, you can raise prices by 10 percent for all dairy products, or you can raise salaries by 5 percent for the people within a certain job category. With an update query, you can change data in existing tables.
- **Append Queries** An append query adds a group of records from one or more tables to the end of one or more tables. For example, suppose that you acquire some new customers and a database containing a table of information on those customers. To avoid typing all this information into your own database, you'd like to append it to your Customers table.
- **Make-Table Queries** A make-table query creates a new table from all or

part of the data in one or more tables. Make-table queries are helpful for creating a table to export to other [Microsoft Access databases](#) or a history table that contains old records.

[SQL queries](#)

An SQL query is a query you create by using an [SQL statement](#). You can use Structured Query Language (SQL) to query, update, and manage relational databases such as Access.

When you create a query in query [Design view](#), Access constructs the equivalent SQL statements behind the scenes for you. In fact, most query properties in the property sheet in query Design view have equivalent clauses and options available in [SQL view](#). If you want, you can view or edit the SQL statement in SQL view. However, after you make changes to a query in SQL view, the query might not be displayed the way it was previously in Design view.

Some SQL queries, called [SQL-specific queries](#), can't be created in the [design grid](#). For [pass-through](#), [data-definition](#), and [union queries](#), you must create the SQL statements directly in SQL view. For [subqueries](#), you enter the SQL in the **Field** row or the **Criteria** row of the query design grid.

[Forms: Putting a friendly face on your data](#)

A form is a type of a database object that is primarily used to enter or display data in a database. You can also use a form as a switchboard that opens other forms and reports in the database, or as a custom dialog box that accepts user input and carries out an action based on the input.



- 1 A data-entry form
- 2 A switchboard form

3 A custom dialog box

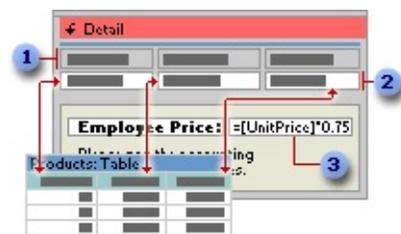
Most forms are bound to one or more tables and [queries](#) in the database. A form's [record source](#) refers to the fields in the underlying tables and queries. A form need not contain all the fields from each of the tables or queries that it is based on.

A bound form stores or retrieves data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the form's design.



- 1 Graphic elements, such as lines and rectangles, are stored in the form's design.
- 2 Data comes from the fields in the underlying record source.
- 3 A calculation comes from an expression, which is stored in the form's design.
- 4 Descriptive text is stored in the form's design.

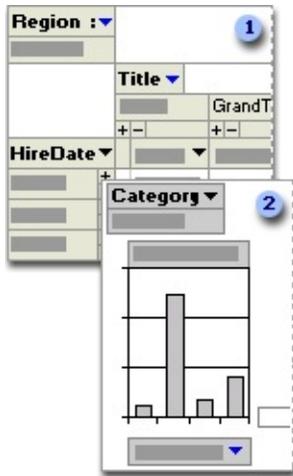
You create a link between a form and its record source by using graphical objects called [controls](#). The most common type of control used to display and enter data is a text box.



- 1 Labels display descriptive text.
- 2 Text boxes display data from and enter data into the Products table.

- 3 A text box uses an expression to calculate a total.

You can also open a form in PivotTable view or PivotChart view to analyze data. In these views, you can dynamically change the layout of a form to present data in different ways. You can rearrange row headings, column headings, and filter fields until you achieve the desired layout. Each time you change the layout, the form immediately recalculates the data based on the new arrangement.



- 1 In PivotTable view, you can view detail or summarized data by arranging fields in the filter, row, column, and detail areas.
- 2 In PivotChart view, you can display data visually by selecting a chart type and viewing data by arranging fields in the filter, series, category, and data areas.

[Creating a form](#)

You can create a form quickly by using the AutoForm command or a wizard. AutoForm creates a form that displays all fields and records in the underlying table or [query](#). A wizard asks you questions and creates a form based on your answers. You can then customize the form the way you want it in [Design view](#).

[Customizing a form](#)

[In Design view](#)

You can customize a form in Design view in the following ways:

Record source Change the tables and queries that a form is based on.

Controlling and assisting the user You can set form properties to allow or prevent users from adding, deleting, or editing records displayed in a form. You can also add custom Help to a form to assist your users with using the form.

Form window You can add or remove **Maximize** and **Minimize** buttons, short cut menus, and other Form window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a form. You can also set section properties to control the appearance and printing of a form.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a form.



[In PivotTable or PivotChart view](#)

You can customize a form in PivotTable or PivotChart view in the following ways:

Add, move, or remove fields You can add fields to the filter, row, column, and detail areas in PivotTable view, and to the filter, category, series, and data areas in PivotChart view. You can also move fields from one area to another and remove fields from the view.

Filter records You can filter data displayed in the view by adding or moving a field to the filter area. You can also filter a field in the row and column area.

Sort records You can sort items in row or column fields in ascending or descending order. You can also sort items in custom order in PivotTable view.

Group records You can group items in row or column fields on intervals, or create custom groups.

Format elements and change captions In PivotTable view, you can change the font settings, alignment, background color, and number format of a field. You can also change the captions of fields and custom groups. In PivotChart view, you can change the chart type, format data markers, and more.

Reports: Presenting your data to the world

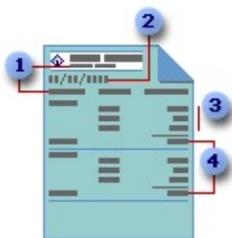
A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.



- 1 Create mailing labels.
- 2 Show totals in a chart.
- 3 Group records into categories.
- 4 Calculate totals.

Most reports are bound to one or more table and [query](#) in the database. A report's record source refers to the fields in the underlying tables and queries. A report need not contain all the fields from each of the tables or queries that it is based on.

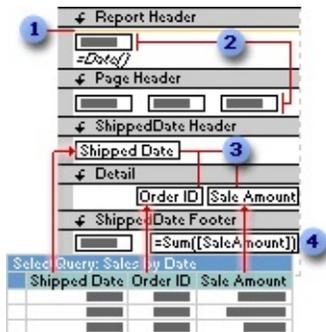
A bound report gets its data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the report's design.



- 1 The report title and column headings are stored in the report's design.
- 2 The date comes from an expression stored in the report's design.

- 3 Data comes from fields in the underlying table, query, or SQL statement.
- 4 Totals come from expressions, which are stored in the report's design.

You create the link between a report and its record source by using graphical objects called [controls](#). Controls can be text boxes that display names and numbers, labels that display titles, or decorative lines that graphically organize the data and make the report more attractive.



- 1 Lines are decorative.
- 2 Labels display descriptive text.
- 3 Text boxes display data from the Sales by Date query.
- 4 A text box uses an expression to calculate a total.

[Creating a report](#)

You can create different types of reports quickly by using wizards. Use the Label Wizard to create mailing labels, the Chart Wizard to create charts, or the Report Wizard to create a standard report. The wizard asks you questions and creates a report based on your answers. You can then customize the report the way you want it in [Design view](#).

[Customizing a report](#)

You can customize a report in the following ways:

Record source Change the tables and queries that a report is based on.

Sorting and grouping data You can sort data in ascending or descending order. You can also group records on one or more fields, and display subtotals and grand totals on a report.

Report window You can add or remove **Maximize** and **Minimize** buttons, change the title bar text, and other Report window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a report. You can also set section properties to control the appearance and printing of a report.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a report.



[Data access pages: Publishing your data to the Web](#)

A data access page is a special type of Web page designed for viewing and working with data from the Internet or an intranet— data that is stored in a [Microsoft Access database](#) or a [Microsoft SQL Server database](#). The data access page may also include data from other sources, such as Microsoft Excel.



[A page in Page view or Microsoft Internet Explorer 5.01 with Service Pack 2 \(SP2\) or later](#)

Using a data access page is similar to using a form: You can view, enter, edit, and delete data in a database. However, you can also use a page outside a [Microsoft Access database](#), so users can update or view data over the Internet or an intranet.

The following illustration shows how the Product data access page, which is used for entering and updating product information, looks in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.



- 1 Enter product information in the text boxes, drop-down list box, and check box.
- 2 Use the record navigation toolbar to navigate, add, delete, save, sort, and filter records, and to get Help.

 [A page in Design view](#)

The following illustration shows the Products data access page in [Design view](#).



- 1 Title in the body of the page
- 2 Group header for the Products group level
- 3 Record navigation section for the Products group level

[Parts of a data access page](#)

The body The body is the basic design surface of a data access page. On a page that supports data entry, you can use it to display informational text, controls bound to data, and sections.

Sections You use sections to display text, data from a database, and toolbars.

Two types of [sections](#) are typically used on pages that support data entry: group header and record navigation sections. A page can also have footer and caption sections.

- **Group header and footer** Used to display data and calculate values.
- **Record navigation** Used to display the [record navigation control](#) for the group level. A record navigation section for a group appears after the group header section. You can't place [bound controls](#) in a record navigation section.
- **Caption** Used to display captions for text boxes and other controls. It appears immediately before the group header. You can't place bound controls in a caption section.

Each group level in a data access page has a [record source](#). The name of the record source is displayed on the [section bar](#) for each section used for a group level.

[Designing different types of data access pages](#)

You design data access pages in [Design view](#) in Microsoft Access. The page is a separate file that is stored outside Access; however, when you create the file, Access automatically adds a shortcut to the file in the Database window. Designing a data access page is similar to designing forms and reports—you use a [field list](#), the [toolbox](#), [controls](#), and so on. However, there are some significant differences in the way that you design and interact with data access pages as opposed to forms and reports. How you design the page depends on what it will be used for.

- **Interactive reporting** This type of data access page is often used to consolidate and group information that is stored in the database, and then publish summaries of the data. For example, a page might publish the sales

performance for each region in which you do business. Using [expand indicators](#), you can go from a general summary of the information, such as a list of all the regions and their combined sales total, to specific details on individual sales within each region. The data access page might provide toolbar buttons for sorting and [filtering](#) the data, as well as for adding, editing, and deleting the data in some or all [group levels](#).

- **Data analysis** This type of data access page may include a [PivotTable list](#), similar to a Microsoft Excel [PivotTable report](#), that lets you reorganize the data to analyze it in different ways. The page might contain a chart that you can use to analyze trends, detect patterns, and compare data in your database. Or it might contain a spreadsheet, in which you can enter and edit data and use formulas to calculate as you do in Excel.

[Using data access pages in Internet Explorer](#)

A data access page is connected directly to a database. When users display the data access page in Internet Explorer, they are viewing their own copy of the page. That means any filtering, sorting, and other changes they make to the way the data is displayed—including changes they make within a [PivotTable list](#) or [spreadsheet](#)—affect only their copy of the data access page. However, changes that they make to the data itself—such as modifying values, and adding or deleting data—are stored in the underlying database, and therefore are available to everyone viewing the data access page.

Users get Help on how to work with the page in Internet Explorer by clicking the **Help**  button on the record navigation toolbar. The Help file that appears is automatically included with any data access page that's published with a record navigation toolbar. If you delete the record navigation toolbar or if you disable its **Help** button, you should provide instructions for using the page to those who will use the page.

Note To view and work with the data access page on the Internet or an intranet, users need Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

[Using data access pages in Microsoft Access](#)

You can also work with a data access page in [Page view](#) in Access. Data access pages can supplement the forms and reports that you use in your database application. When deciding whether to design a data access page, form, or

report, consider the tasks that you want to perform.

You get Help on how to work with the page in Page view by clicking the **Help**  button on the **Page View** toolbar to display Access Help. Clicking the **Help** button on a record navigation toolbar on the page displays Help for working with the page in Microsoft Internet Explorer. As mentioned previously, this Help file is automatically included with any data access page that's published with a record navigation toolbar. You can delete the **Help** button on pages that are used only within Access, or you can modify the button to provide custom Help.

 [Where a data access page gets its data](#)

 [Types of data sources for a data access page](#)

A data access page gets its data from a [Microsoft Access database](#), or a [Microsoft SQL Server database](#) version 6.5 or later. To design a page using data from one of these databases, the page must be connected to the database. If you have already opened an Access database or an [Access project](#) that's connected to a SQL Server database, the data access page you create automatically connects to the current database and stores that path in the **ConnectionString** property of the data access page. When a user browses to the page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later or displays the page in [Page view](#), it displays current data from the underlying database by using the path defined in the **ConnectionString** property. If you design the data access page while the database is on a local drive, Access will use the local path, which means the data won't be accessible to others. For this reason, it's important that you move or copy the database to a network location that's accessible to users of your page. Once the database is on the network share, open the database using the [UNC](#) address. If you move or copy the database after you've already designed the page, you'll have to update the path in the **ConnectionString** property to point to the new location.

Instead of having to update the **ConnectionString** property of each page in the database separately, you can choose to create a connection file. A connection file stores the connection information for a data access page, and can be shared between multiple data access pages. When you open a data access page that uses a connection file, the page reads the connection file and connects to the appropriate database. After creating the connection file, if you move or copy the database, all you have to do is edit the connection information in the connection

file.

[Types of data sources for Microsoft Office Web Components on a data access page](#)

Although the data access page gets its data from a [Microsoft Access database](#) or [Microsoft SQL Server database](#), [Microsoft Office Web Component](#) controls on the page can display data from either of these databases or from other sources. For example, the page may contain a [PivotTable list](#), [spreadsheet](#), or [chart](#) with data from a Microsoft Excel worksheet or a database other than Access or SQL Server. Depending on the data source, the data in these controls may be a snapshot of the original data or it may be live data displayed through a separate connection that the control has with its data source.

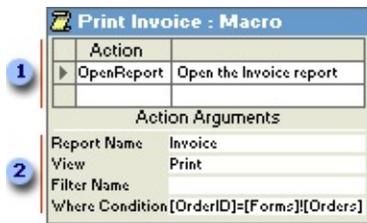
To create an Office Web Component with data from a source other than the current database, you might have to use a program other than Access to first publish the data to a Web page. For example, to create a PivotTable list that contains data from an Excel worksheet, you have to publish the data from Excel. Then, in Access, you can open the Web page and turn it into a data access page by adding controls that are bound to an Access or SQL Server database. You can further customize the page by adding labels, pictures, or other features to enhance the page's appearance. From within Access, you can create a PivotTable list that uses data from an external source other than Excel.

[Macros: Putting your data to work](#)

Macros are a set of actions that you can create to help you to automate common tasks. By using groups of macros, you can perform several tasks at once.

[What is a macro?](#)

A macro is a set of one or more [actions](#) that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.



- 1 When you create a macro, you enter the actions you want to carry out in this portion of the [Macro window](#).
- 2 You can specify arguments for an action in this portion of the window.

A macro can be one macro composed of a sequence of actions, or it can be a [macro group](#). You can also use a [conditional expression](#) to determine whether in some cases an action will be carried out when a macro runs.

The following macro is composed of a series of actions. Microsoft Access carries out these actions each time the macro runs. To run this macro, you refer to the macro name Review Products.

Review Products : Macro	
Action	Comment
	Attached to the Add
Echo	Freeze screen while
OpenForm	Open the Product List
MoveSize	Position the Product L

[What is a macro group?](#)

If you have numerous macros, grouping related macros in [macro groups](#) can help you to manage your database more easily.

For example, the following macro group, named Buttons, is made up of three related macros: Employees, Products, and Reps. Each macro carries out the OpenForm action, and the Products macro also carries out the MoveSize action.

Buttons : Macro	
Macro Name	Action
Employees	OpenForm
Products	OpenForm
	MoveSize
Reps	OpenForm

The name in the **Macro Name** column identifies each macro. When you run a macro in a macro group, Microsoft Access carries out the action in the action

column and any actions that immediately follow with a blank **Macro Name** column.

You can run a macro in a macro group in an event or [event procedure](#) by typing the macro group name followed by a period and then the macro name. In the preceding example, to refer to the Employees macro in the Buttons macro group, you would type **Buttons.Employees**.

[Conditional actions](#)

In some cases, you may want to carry out an action or series of actions in a macro only if a particular [condition](#) is true. For example, if you're using a macro to validate data in a form, you might want to display one message in response to one set of values entered in a record and another message in response to a different set of values. In cases like these, you can use conditions to control the flow of the macro.

A condition is a logical [expression](#) that evaluates to True/False or Yes/No. The macro follows different paths depending on whether the condition is true or false.

When you run the macro, Microsoft Access evaluates the first [conditional expression](#). If the condition is true, Microsoft Access carries out the action in that row and any of the immediately following actions that are preceded by an ellipsis (...) in the **Condition** column.

Microsoft Access then runs any additional actions in the macro that have a blank **Condition** column until it reaches another expression, a macro name, or the end of the macro.

If the condition is false, Microsoft Access ignores the action and any immediately following actions that are preceded by an ellipsis in the **Condition** column and moves to the next [action row](#) that contains another condition or a blank **Condition** column.

The following macro runs the MsgBox and the StopMacro actions only when the expression in the **Condition** column is true (when there is a [Null](#) value in the SupplierID field).

Review Products : Macro	
Condition	Action
IsNull([SupplierID])	MsgBox StopMacro

[Action arguments](#)

Action arguments are additional information required by some macro actions — for example, the object affected by the action or special conditions under which the action is carried out. After you add an action to a macro, you set the arguments for the action in the lower portion of the [Macro window](#). These arguments give Microsoft Access additional information on how to carry out the action.

Tips for setting action arguments:

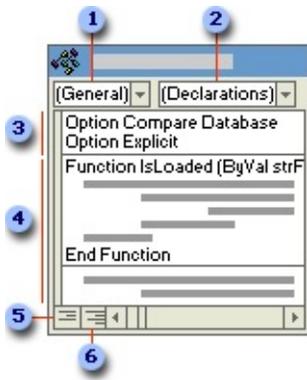
- In general, it's a good idea to set action arguments in the order they're listed, because choices for one argument may determine those for arguments that follow.
- If you add an action to your macro by dragging a [database object](#) from the [Database window](#), Microsoft Access automatically sets appropriate arguments for that action.
- If an action has an argument that calls for the name of a database object, you can set the argument and the corresponding object type argument automatically by dragging the object from the Database window to the argument box.
- You can use an [expression](#) preceded by an equal sign (=) to set many action arguments.

[Modules: Automate routine tasks and create business solutions](#)

A module is essentially a collection of declarations, statements, and procedures stored together as one named unit to organize your [Microsoft Visual Basic](#) code. Microsoft Access has two types of modules: [standard modules](#) and [class modules](#).

[What is a module?](#)

A module is a collection of Visual Basic declarations and procedures that are stored together as a unit.



- 1 **Object** box
- 2 **Procedure** box
- 3 Declarations
- 4 Procedures
- 5 **Procedure View** button
- 6 **Full Module View** button

[Class modules](#)

[Form](#) and [report modules](#) are class modules that are associated with a particular form or report. Form and report modules often contain [event procedures](#) that run in response to an event on the form or report. You can use event procedures to control the behavior of your forms and reports, and their response to user actions, such as clicking a command button.

When you create the first event procedure for a form or report, Microsoft Access automatically creates an associated form or report module.



- 1 To view or add a form procedure, choose one of the form's objects from the **Object** box ...

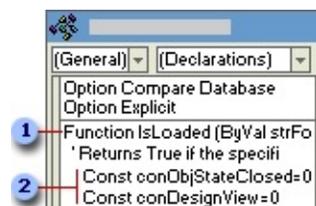
2 ... and then choose an event from the **Procedure** box. Names of events that already have procedures are bold.

Procedures in your form and report modules can call procedures you have added to standard modules.

In Access 97 or later, class modules can also exist independent of a form or report, and this type of class module is listed in the [Database window](#). You can use a class module to create a definition for a custom object. In Access 95, class modules exist in association only with a form or report.

[Standard modules](#)

Standard modules contain general procedures that aren't associated with any other object, and frequently used procedures that can be run from anywhere within your database. The major difference between a standard module and a class module that is not associated with a particular object is one of scope and lifetime. The value of any variables or constants that are declared or exist in a class module without an associated object are available for use only while that code is running and only from that object.



- 1 Function name and arguments
- 2 Visual Basic declarations and code statements

Standard modules are listed in **Modules** under **Objects** in the [Database window](#). Form, report, and standard modules are also listed in the [Object Browser](#).

[Relationships in a database](#)

After you've set up different tables for each subject in your [Microsoft Access database](#), you need a way of telling Microsoft Access how to bring that information back together again. The first step in this process is to define [relationships](#) between your tables. After you've done that, you can create queries,

forms, and reports to display information from several tables at once. For example, this form includes information from four tables:

The screenshot shows an 'Orders' form with the following data:

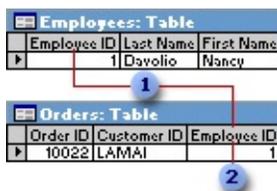
- Customers table:** Bill To: Maison Dewey (callout 1)
- Orders table:** Order ID: 10523, Order Date: 07-Jun-00 (callout 2)
- Products table:** Product list with columns Product, Unit Price, and Quantity (callout 3)
- Order Details table:** Grid showing product details for the order (callout 4)

Product	Unit Price	Quantity
Gudbrandsbros	\$24.00	14
Scottish Longbread	\$12.50	20
Pâté chinois	\$36.00	10

- 1 The Customers table
- 2 The Orders table
- 3 The Products table
- 4 The Order Details table

[How relationships work](#)

In the previous example, the fields in four tables must be coordinated so that they show information about the same order. This coordination is accomplished with relationships between tables. A relationship works by matching data in key fields— usually a field with the same name in both tables. In most cases, these matching fields are the [primary key](#) from one table, which provides a unique identifier for each record, and a [foreign key](#) in the other table. For example, employees can be associated with orders they're responsible for by creating a relationship between the EmployeeID fields.



- 1 EmployeeID appears in both tables— -as a primary key ...
- 2 ... and as a foreign key.

A one-to-many relationship

A one-to-many relationship is the most common type of relationship. In a one-to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.

Suppliers: Table	
Supplier ID	Company Name
1	Exotic Liquids
2	New Orleans Cajun Delights
3	Grandma Kelly's Homestead
4	Tokyo Traders

Products: Table		
Product ID	Product Name	Supplier ID
1	Chai	1
2	Chang	1
3	Aniseed Syrup	1
4	Chef Anton's Cajun Seasoning	2
5	Chef Anton's Gumbo Mix	2

- 1 One supplier ...
- 2 ... can supply more than one product ...
- 3 ... but each product has only one supplier.

A many-to-many relationship

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose [primary key](#) consists of two fields—the [foreign keys](#) from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table. For example, the Orders table and the Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.

Orders: Table	
Order ID	Customer ID
10000	FRANS
10001	MEREP
10002	FOLKO

Order Details: Table		
Order ID	Product ID	
10001	25	
10001	40	
10001	53	
10002	25	

Products: Table	
Product ID	Product Name
25	NuNuCa Nuß-Nougat Creme
26	Gumbär Gummibärchen

- 1 Primary key from the Orders table
- 2 Primary key from the Products table
- 3 One order can have many products ...
- 4 ... and each product can appear on many orders.

[A one-to-one relationship](#)

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. For example, you might want to create a table to track employees participating in a fundraising soccer game. Each soccer player in the Soccer Players table has one matching record in the Employees table.

Employees: Table		
Employee ID	Last Name	First Name
1	Davolio	Nancy
2	Fuller	Andrew
3	Leverling	Janet
4	Peacock	Margaret
5	Buchanan	Steven

Soccer Players: Table		
Employee ID	Player Nickname	Skill Level
1	Slammin' Man	2
3	Ace	1
5	Stevemeister	2

- 1 Each soccer player has one matching record in the Employees table.
- 2 This set of values is a subset of the EmployeeID field and the Employees table.

[About defining relationships](#)

The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

- A one-to-many relationship is created if only one of the related fields is a [primary key](#) or has a [unique index](#).
- A one-to-one relationship is created if both of the related fields are primary

keys or have unique indexes.

- A many-to-many relationship is really two one-to-many relationships with a third table whose primary key consists of two fields— the [foreign keys](#) from the two other tables.

You can also create a relationship between a table and itself. This is useful in situations where you need to perform a Lookup within the same table. In the Employees table, for example, you can define a relationship between the EmployeeID and ReportsTo fields, so that the ReportsTo field can display employee data from a matching EmployeeID.

Note If you drag a field that isn't a primary key and doesn't have a unique index to another field that isn't a primary key and doesn't have a unique index, an indeterminate relationship is created. In queries containing tables with an indeterminate relationship, Microsoft Access displays a default [join](#) line between the tables, but [referential integrity](#) won't be enforced, and there's no guarantee that records are unique in either table.

[Referential integrity](#)

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. You can set referential integrity when all of the following conditions are met:

- The matching field from the [primary table](#) is a [primary key](#) or has a [unique index](#).
- The related fields have the same [data type](#). There are two exceptions. An [AutoNumber](#) field can be related to a Number field with a **FieldSize** property setting of **Long Integer**, and an AutoNumber field with a **FieldSize** property setting of **Replication ID** can be related to a Number field with a **FieldSize** property setting of **Replication ID**.
- Both tables belong to the same Microsoft Access database. If the tables are linked tables, they must be tables in Microsoft Access format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

The following rules apply when you use referential integrity:

- You can't enter a value in the [foreign key](#) field of the related table that doesn't exist in the primary key of the primary table. However, you can enter a **Null** value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.
- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

[Cascading updates and deletes](#)

For relationships in which [referential integrity](#) is enforced, you can specify whether you want Microsoft Access to automatically [cascade update](#) and [cascade delete](#) related records. If you set these options, delete and update operations that would normally be prevented by referential integrity rules are allowed. When you delete records or change [primary key](#) values in a [primary table](#), Microsoft Access makes necessary changes to related tables to preserve referential integrity.

If you select the **Cascade Update Related Fields** check box when defining a relationship, any time you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship isn't broken. Microsoft Access cascades updates without displaying any message.

Note If the primary key in the primary table is an [AutoNumber](#) field, setting the **Cascade Update Related Fields** check box will have no effect, because you can't change the value in an AutoNumber field.

If you select the **Cascade Delete Related Records** check box when defining a relationship, any time you delete records in the primary table, Microsoft Access

automatically deletes related records in the related table. For example, if you delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the **Cascade Delete Related Records** check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a [delete query](#), Microsoft Access automatically deletes the records in related tables without displaying a warning.

Access Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#)

[N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

A

[Back to top](#)

Absolute Positioning

Places the element relative to either the element's parent or, if there isn't one, the body. Values for the element's **Left** and **Top** properties are relative to the upper-left corner of the element's parent.

Access Project

An Access file that connects to a Microsoft SQL Server database and is used to create client/server applications. A project file doesn't contain any data or data-definition-based objects such as tables and views.

Action

The basic building block of a macro; a self-contained instruction that can be combined with other actions to automate tasks. This is sometimes called a command in other macro languages.

Action Argument

Additional information required by some macro actions. For example, the object affected by the action or special conditions under which the action is carried out.

Action List

The list that appears when you click the arrow in the **Action** column of the Macro window.

Action Query

A query that copies or changes data. Action queries include append, delete, make-table, and update queries. They are identified by an exclamation point (!) next to their name in the Database window.

Action Row

A row in the upper part of the Macro window in which you enter macro names, actions, conditions, and comments associated with a particular macro or macro group.

ADE File

A Microsoft Access project (.adp) file with all modules compiled and all editable source code removed.

Admin Account

The default user account. When you install Access, the Setup program automatically includes the Admin user account in the workgroup information file it creates.

Admins Group

The system administrator's group account that retains full permissions on all databases used by a workgroup. The Setup program automatically adds the default Admin user account to the Admins group.

Advanced Filter Sort

A window in which you can create a filter from scratch. You enter criteria expressions in the filter design grid to restrict the records in the open form or datasheet to a subset of records that meet the criteria.

Aggregate Function

A function, such as **Sum**, **Count**, **Avg**, or **Var**, that you use to calculate totals.

Anonymous Replica

In an Access database, a special type of replica in which you don't keep track of individual users. The anonymous replica is particularly useful in an Internet situation where you expect many users to download replicas.

ANSI SQL Query Mode

One of two types of SQL syntax: ANSI-89 SQL (also called Microsoft Jet SQL and ANSI SQL), which is the traditional Jet SQL syntax; and ANSI-92 SQL, which has new and different reserved words, syntax rules, and wildcard characters.

Append Query

An action query that adds the records in a query's result set to the end of an existing table.

Application Background

The background area of an application window. In Access, this is the screen area behind the Database window or any other open windows, but contained within the Access application window.

Attach Link

An action that establishes a connection to data from another application so that you can view and edit the data in both the original application and in Access.

Attached Table

A table stored in a file outside the open database from which Access can access records. You can add, delete, and edit records in a linked table, but you can't change its structure.

Auto format

A collection of formats that determines the appearance of the controls and sections in a form or report.

Auto Number

In a Microsoft Access database, a field data type that automatically stores a unique number for each record as it's added to a table. Three kinds of numbers can be generated: sequential, random, and Replication ID.

Autofiltering

Filtering data in PivotTable or PivotChart view by selecting one or more items in a field that allows filtering.

Automatic Link

A link from an OLE object in Access to an OLE server that automatically updates the object in Access when the information in the object file changes.

B

[Back to top](#)

Bigint Data Type

In an Access project, a data type of 8 bytes (64 bits) that stores whole numbers in the range of -2^{63} (-9,223,372,036,854,775,808) through $2^{63}-1$ (9,223,372,036,854,775,807).

Binary Data Type

In an Access project, a fixed-length data type with a maximum of 8,000 bytes of binary data.

Bit Data Type

In an Access project, a data type that stores either a 1 or 0 value. Integer values other than 1 or 0 are accepted, but are always interpreted as 1.

Bit Mask

A value used with bit-wise operators (**And**, **Eqv**, **Imp**, **Not**, **Or**, and **Xor**) to test, set, or reset the state of individual bits in a bit-wise field value.

Body

The basic design surface of a data access page. It displays informational text, controls bound to data, and sections.

Bookmark

A property of a **Recordset** object or a form that contains a binary string identifying the current record.

Bound Column

The column in a list box, combo box, or drop-down list box that's bound to the field specified by the control's **ControlSource** property.

Bound Field

A control used on a form, report, or data access page to display or modify data from a table, query, or SQL statement. The control's **ControlSource** property stores the field name to which the control is bound.

Bound HTML Control

A control used on a data access page to bind HTML code to a Text or Memo field in an Access database or to a text, ntext, or varchar column in an Access project. You cannot edit the contents of a bound span control.

Bound Hyperlink Control

A control used on a data access page to bind a link, an intranet address, or an Internet address to an underlying Text field. You can click the hyperlink to navigate to the target location.

Bound Object Frame

A control on a form or report used to display and manipulate OLE objects that are stored in tables.

Bound Picture

A control used on a form, report, or data access page to bind an image to an OLE Object field in an Access database or an image column in an Access project.

Builder

An Access tool that simplifies a task. For example, you can quickly create a complex expression with the Expression Builder.

Built-In Toolbar

A toolbar that is part of Access when it's installed on your computer. In contrast, a custom toolbar is one you create for your own application.

Byte Data Type

An Access database data type used to hold small positive integers ranging from 0 to 255.

C

[Back to top](#)

Calculated Control

A control used on a form, report, or data access page to display the result of an expression. The result is recalculated each time there is a change in any of the values that the expression is based on.

Calculated Field

A field, defined in a query, that displays the result of an expression rather than displaying stored data. The value is recalculated each time a value in the expression changes.

Call Tree

All modules that might be called by any procedure in the module in which code is currently running.

Caption Section

The section on a grouped data access page that displays captions for columns of data. It appears immediately before the group header. You cannot add a bound control to a caption section.

Cascade Delete

For relationships that enforce referential integrity between tables, the deletion of all related records in the related table or tables when a record in the primary table is deleted.

Cascade Update

For relationships that enforce referential integrity between tables, the updating of all related records in the related table or tables when a record in the primary table is changed.

Cascading Event

A sequence of events caused by an event procedure directly or indirectly calling itself; also called an event cascade or a recursion. Be careful using cascading events, because they often result in stack-overflow or other run-time errors.

Category Field

A field displayed in the category area of PivotChart view. Items in a category field appear as labels on the category axis.

Channel Number

An integer that corresponds to an open Dynamic Data Exchange (DDE) channel. Channel numbers are assigned by Microsoft Windows 95 or later, created with the **DDEInitiate** function, and used by other DDE functions and statements.

Char Data Type

In an Access project, a fixed-length data type with a maximum of 8,000 ANSI characters.

Check Box

A control that indicates whether an option is selected. A check mark appears in the box when the option is selected.

Class Module

A module that can contain the definition for a new object. Each instance of a class creates a new object. Procedures defined in the module become properties and methods of the object. Class modules can exist alone or with forms and reports.

Class Name

The name used to refer to a class module. If the class module is a form or report module, the class name is prefaced with the type of module. For example, Form_OrderForm.

Class Name OLE

A predefined name, used to refer to an OLE object in Visual Basic. It consists of the name of the application used to create the OLE object, the object's type, and, optionally, the version number of the application.

Example: Excel.Sheet

Code Stub

A segment of Visual Basic code that defines the beginning and end of a procedure.

Column

A location within a database table that stores a particular type of data. It is also the visual representation of a field in a datasheet, and in an Access database, the query design grid or the filter design grid.

Column Area

The part of PivotTable view that contains column fields.

Column Field

A field in the column area of PivotTable view. Items in column fields are listed across the top of a PivotTable list. Inner column fields are closest to the detail area; outer column fields are above the inner column fields.

Column Selector

The horizontal bar at the top of a column. You can click a column selector to select an entire column in the query design grid or the filter design grid.

Combo Box

A control used on a form that provides the combined functionality of a list box and a text box. You can type a value, or click the control to display a list and select an item from the list.

Command Button

A control that runs a macro, calls a Visual Basic function, or runs an event procedure. This is sometimes called a push button in other applications.

Comparison Operator

An operator used to compare two values or expressions; For example, < (less than), > (greater than), and = (equal to).

Compound Control

A control and an attached label, such as a text box with an attached label.

Conditional Filtering

Filtering a field to show the top or bottom n items based on a total. For example, you could filter for the three cities that generated the most sales or the five products that are least profitable.

Conditional Formatting

Formatting the contents of a control in a form or report based on one or more conditions. A condition can reference another control, the control with focus, or a user-defined Visual Basic for Applications function.

Conflict

Can occur if data has changed in the same record of two replica set members. When a conflict occurs, a winning change is selected and applied in all replicas, and the losing change is recorded as a conflict at all replicas.

Connection String

A string expression used to open an external database.

Constraint

A restriction placed upon the value that can be entered into a column or a row. For example, values in the Age column cannot be less than 0 or greater than 110.

Container Application

An application that contains a linked or embedded OLE object from another application. For example, if an OLE object in an Access database contains a Microsoft Excel worksheet, Access is the OLE container.

Continuous Form

A form that displays more than one record on the screen in Form view.

Control Containing Hyperlink

A control that makes it possible for a user to jump to a document, Web page, or object. For example, a text box bound to a field that contains hyperlinks.

Crosstab Query

A query that calculates a sum, average, count, or other type of total on records, and then groups the result by two types of information: one down the left side of the datasheet and the other across the top.

Currency Data Type

In a Microsoft Access database, a data type that is useful for calculations

involving money or for fixed-point calculations in which accuracy is extremely important.

Current Record

The record in a recordset that you can modify or retrieve data from. There is only one current record in a recordset, but a recordset may have no current record, for example, after a record has been deleted from a dynaset-type recordset.

Cursor Data Type

In an Access project, a data type you can only use for creating a cursor variable. This data type cannot be used for columns in a table. A cursor is a mechanism used to work with one row at a time in the result set of a SELECT statement.

Custom Group

An item of a custom group field. A custom group contains two or more items from a row or column field.

Custom GroupField

A field in the row or column area that contains custom groups as its items.

Custom Order

User-defined sort order. For example, you could define a custom sort order to display values in the EmployeeTitle column on the basis of the title's seniority.

Custom Properties Dialog

A custom property sheet that allows users to set properties for an ActiveX control.

Custom Toolbar

A toolbar that you create for your application. In contrast, a built-in toolbar is part of Access when it's installed on your computer.

D

[Back to top](#)

Data Access Page Properties

Attributes of a data access page that identify the database the page is connected to and define the page's appearance and behavior.

Data Access Page Window

The window that displays data access pages in Design view and Page view.

Data Area

The part of PivotTable or PivotChart view that contains summary data.

Values in the data area are displayed as records in PivotTable view and as data points in PivotChart view.

Data Definition

The fields in underlying tables and queries, and the expressions, that make up the record source for a data access page.

Data Field

A field that contains summarized data in PivotTable or PivotChart view. A data field usually contains numeric data.

Data Files

An Access database or Access project file. An Access database stores database objects and data in an .mdb file. A project file doesn't contain data and is used to connect to a Microsoft SQL Server database.

Data Item

An application-specific piece of data that can be transferred over a DDE channel.

Data Label

A label that provides additional information about a data marker, which represents a single data point or value.

Data Marker

A bar, area, dot, slice, or other symbol in a chart that represents a single data point or value. Related data markers in a chart constitute a data series.

Data Outline

A window that you can display when a data access page is open in Design view. The data outline provides a tree view of the data model of a page. You can select and set properties of the elements displayed in the window.

Data Series

Related data points that are plotted in a chart. Each data series in a chart has a unique color or pattern. You can plot one or more data series in a chart.

Data Source Control

The engine behind data access pages and Microsoft Office Web Components that manages the connection to the underlying data source. The data source control has no visual representation.

Database Diagram

A graphical representation of any portion of a database schema. It can be either a whole or partial picture of the structure of the database. It includes tables, the columns they contain, and the relationships between the tables.

Database Objects

An Access database contains objects such as tables, queries, forms, reports, pages, macros, and modules. An Access project contains objects such as

forms, reports, pages, macros, and modules.

Database Replication

The process of creating two or more special copies (replicas) of an Access database. Replicas can be synchronized changes made to data in one replica, or design changes made in the Design Master, are sent to other replicas.

Database Window

The window that appears when you open an Access database or an Access project. It displays shortcuts for creating new database objects and opening existing objects.

Data-definition Query

A SQL-specific query that contains data definition language (DDL) statements. These statements allow you to create or alter objects in the database.

Datasheet

Data from a table, form, query, view, or stored procedure displayed in a row-and-column format.

Datasheet View

A window that displays data from a table, form, query, view, or stored procedure in a row-and-column format. In Datasheet view, you can edit fields, add and delete data, and search for data.

Date Data Type

An Access database data type used to hold date and time information.

Datetime Data Type

In an Access project, a date and time data type that ranges from January 1, 1753, to December 31, 9999, to an accuracy of three-hundredths of a second, or 3.33 milliseconds.

Decimal Data Type

An exact numeric data type that holds values from $-10^{38} - 1$ through $10^{38} - 1$. You can specify the scale (maximum total number of digits) and precision (maximum number of digits to the right of the decimal point).

Decimal Data Type Microsoft Access Database

An exact numeric data type that holds values from $-10^{28} - 1$ through $10^{28} - 1$. You can specify the scale (maximum number of digits) and precision (maximum total number of digits to the right of the decimal point).

Declarations Section

The section of a module containing declarations that apply to every procedure in the module. It can include declarations for variables,

constants, user-defined data types, and external procedures in a dynamic-link library.

Default Property

A property that you can set for a control so that each time a new control of that type is created, this property will have the same value.

Default Value

A value that is automatically entered in a field or control when you add a new record. You can either accept the default value or override it by typing a value.

Delete Query

A query (SQL statement) that removes rows matching the criteria you specify from one or more tables.

Design Master

The only member of the replica set in which you can make changes to the database structure that can be propagated to other replicas.

Design View

A window that shows the design of these database objects: tables, queries, forms, reports, macros, and data access pages. In Design view, you can create new database objects and modify the design of existing ones.

Detail Area

The part of PivotTable view that contains detail and total fields.

Detail Field

A field that displays all rows, or records, from the underlying record source.

Detail Section

Used to contain the main body of a form or report. This section usually contains controls bound to the fields in the record source, but can also contain unbound controls, such as labels that identify a field's contents.

Direct Synchronization

A method used to synchronize data between replicas that are connected directly to the local area network and are available through shared network folders.

Disabled Field

A control that appears dimmed on a form or data access page. A disabled control cannot get the focus and will not respond to mouse clicks.

DLL

A set of routines that can be called from Visual Basic procedures and are loaded and linked into your application at run time.

Document Properties

Properties, such as title, subject, and author, that are stored with each data

access page.

Domain

A set of records defined by a table, a query, or an SQL expression. Domain aggregate functions return statistical information about a specific domain or set of records.

Domain Function

A function, such as **DAvg** or **DMax**, that is used to calculate statistics over a set of records (a domain).

Drop Area

An area in PivotTable or PivotChart view where you can drop fields from the field list to display the data in the field. The labels on each drop area indicate the types of fields you can create in the view.

Drop-down List Box

A control on a data access page that, when clicked, displays a list from which you can select a value. You cannot type a value in a drop-down list box.

E

[Back to top](#)

Echo

The process of Access updating or repainting the screen while a macro is running.

Edit Control

Also known as a text box, an edit control is a rectangular window in which a user can enter and edit text.

Embed

To insert a copy of an OLE object from another application. The source of the object, called the OLE server, can be any application that supports object linking and embedding. Changes to an embedded object are not reflected in the original object.

Enabled Database

A previous-version database that has been opened in Access 2000 or later without converting its format. To change the design of the database, you must open it in the version of Access in which it was created.

Exclusive

A type of access to data in a database shared over a network. When you open a database in exclusive mode, you prevent others from opening the

database.

Expand Control

A control on a data access page that, when clicked, expands or collapses a grouped record to display or hide its detail records.

Expand Indicator

A button that is used to expand or collapse groups of records; it displays the plus (+) or minus (-) sign.

Export

To output data and database objects to another database, spreadsheet, or file format so another database or program can use the data or database objects. You can export data to a variety of supported databases, programs, and file formats.

Expression Builder

An Access tool you can use to create an expression. It includes a list of common expressions that you can select.

External Database

The source of the table that is to be linked or imported to the current database, or the destination of a table that is to be exported.

External Table

A table outside the open Access database or Access project.

F

[Back to top](#)

Field Data Type

A characteristic of a field that determines what kind of data it can store. For example, a field whose data type is Text can store data consisting of either text or number characters, but a Number field can only store numerical data.

Field List

A window that lists all the fields in the underlying record source or database object, except in data access page Design view. In data access page Design view, it lists all the record sources and their fields in the underlying database.

Field Selector

A small box or bar that you click to select an entire column in a datasheet.

Fill

A report magnification that fills the Report Snapshot window by fitting

either the width or the height of a page, depending on whether the report is in portrait or landscape orientation.

Filter

A set of criteria applied to data in order to display a subset of the data or to sort the data. In Access, you can use filtering techniques, such as Filter By Selection and Filter By Form to filter data.

Filter Area

The part of PivotTable or PivotChart view that contains filter fields.

Filter By Form

A technique for filtering data that uses a version of the current form or datasheet with empty fields in which you can type the values you want the filtered records to contain.

Filter By Selection

A technique for filtering records in a form, datasheet, or data access page in which you retrieve only records that contain the selected value.

Filter Excluding Selection

A technique in which you filter records in a form, datasheet, or data access page to retrieve only those records that don't contain the selected value.

Filter Field

A field in the filter area that you can use to filter data displayed in PivotTable or PivotChart view. Filter fields perform the same functions as page fields in Microsoft Excel PivotTable reports.

Filter For Input

A technique for filtering records that uses a value or expression you enter to find only records that contain the value or satisfy the expression.

Fixed-width Text

A file containing data where each field has a fixed width.

Float Data Type

In an Access project, an approximate numeric data type with 15-digit precision. It can hold positive values from approximately $2.23E - 308$ through $1.79E + 308$, negative values from approximately $-2.23E - 308$ through $-1.79E + 308$, or zero.

Floating Toolbar

Able to move freely as its own window. A floating window is always on top. Toolbars, menu bars, the toolbox, and palettes can float.

Foreign Key

One or more table fields (columns) that refer to the primary key field or fields in another table. A foreign key indicates how the tables are related.

Foreign Table

A table (such as Customer Orders) that contains a foreign key field (such as CustomerID) that's the primary key field in another table (such as Customers) in the database, and that is usually on the "many" side of a one-to-many relationship.

Form

An Access database object on which you place controls for taking actions or for entering, displaying, and editing data in fields.

Form Footer

Use to display instructions for using a form, command buttons, or unbound controls to accept input. Displays at the bottom of the form in Form view and at the end of a printout.

Form Header

Use to display a title for a form, instructions for using it, or command buttons that open related forms or carry out other tasks. Displays at the top of the form in Form view and at the beginning of a printout.

Form Module

A module that includes code for all event procedures triggered by events occurring on a specific form or its controls.

Form Properties

Attributes of a form that affect its appearance or behavior. For example, the **DefaultView** property is a form property that determines whether a form will automatically open in Form view or Datasheet view.

Form Selector

The box where the rulers meet, in the upper-left corner of a form in Design view. Use the box to perform form-level operations, such as selecting the form.

Form View

A window that displays a form to show or accept data. Form view is the primary means of adding and modifying data in tables. You can also change the design of a form in this view.

Form Window

A window in which you work with forms in Design view, Form view, Datasheet view, or Print Preview.

Format

Specifies how data is displayed and printed. An Access database provides standard formats for specific data types, as does an Access project for the equivalent SQL data types. You can also create custom formats.

Front-end/Back-end App

An application consisting of a "back-end" database file that contains tables,

and copies of a "front-end" database file that contain all other database objects with links to the "back-end" tables.

Function

A query that takes input parameters and returns a result like a stored procedure. Types: scalar (multistatement; returns one value), inline (one statement; an updateable table value), and table (multistatement; table value).

Function Procedure

A procedure that returns a value and that can be used in an expression. You declare a function with the **Function** statement and end it with the **End Function** statement.

G

[Back to top](#)

General Sort Order

The default sort order determines how characters are sorted in the entire database, such as in tables, queries, and reports. Define General sort order if you plan to use a database with multiple language editions of Access.

Global Menu Bar

A special custom menu bar that replaces the built-in menu bar in all windows in your application, except where you've specified a custom menu bar for a form or report.

Global Replica

A replica in which changes are fully tracked and can be exchanged with any global replica in the set. A global replica can also exchange changes with any local or anonymous replicas for which it becomes the hub.

Global Shortcut Menu

A custom shortcut menu that replaces the built-in shortcut menu for the following objects: fields in table and query datasheets; forms and form controls in Form view, Datasheet view, and Print Preview; and reports in Print Preview.

Global Variable

A variable you declare with the **Public** keyword in the Declarations section of a module. A public variable can be shared by all the procedures in every module in a database.

Graph

A graphical representation of data in a form, report, or data access page.

Grid

An arrangement of vertical and horizontal dotted and solid lines that help you position controls precisely when you design a form, report, or data access page.

Grid Datasheet View

Vertical and horizontal lines that visually divide rows and columns of data into cells in a table, query, form, view, or stored procedure. You can show and hide these grid lines.

Group

Two or more controls that can be treated as one unit while designing a form or report. You can select the group instead of selecting each individual control as you're arranging controls or assigning properties.

Group Account

A collection of user accounts in a workgroup, identified by group name and personal ID (PID). Permissions assigned to a group apply to all users in the group.

Group Filter Control

A drop-down list box control on a data access page that retrieves records from an underlying recordset based on the value you select from the list. On a grouped page, the control retrieves a specific group of records.

Group Footer

Use to place information, such as group name or group total, at the end of a group of records.

Group Header

Use to place information, such as group name or group total, at the beginning of a group of records.

Group Level

The depth at which a group in a report or data access page is nested inside other groups. Groups are nested when a set of records is grouped by more than one field, expression, or group record source.

Grouped Data Access Page

A data access page that has two or more group levels.

GUID

A 16-byte field used in an Access database to establish a unique identifier for replication. GUIDs are used to identify replicas, replica sets, tables, records, and other objects. In an Access database, GUIDs are referred to as Replication IDs.

H

[Back to top](#)

Handle

A unique **Integer** value (**Long** in a 32-bit environment) you use to identify and access a Form or Report window. Use the **hWnd** property to return the handle.

Hotspot

A control used on a data access page to display an unbound image that represents a hyperlink to a file or Web page. In Browse mode, you can click the image to navigate to the target location.

Hub

A global replica that all replicas in the replica set synchronize their changes to. The hub serves as the parent replica.

Hyperlink Address

The path to a destination such as an object, document, or Web page. A hyperlink address can be a URL (address to an Internet or intranet site) or a UNC network path (address to a file on a local area network).

Hyperlink Data Type

A data type for an Access database field that stores hyperlink addresses. An address can have up to four parts and is written using the following format: *displaytext#address#subaddress#*.

Hyperlink Field

A field that stores hyperlink addresses. In an Access database (.mdb), it's a field with a Hyperlink data type. In an Access project (.adp), it's a field that has the **IsHyperlink** property set to **True**.

I

[Back to top](#)

IDC HTX files

Microsoft Internet Information Server uses an IDC file and an HTX file to retrieve data from an ODBC data source and format it as an HTML document.

Identifier

An element of an expression that refers to the value of a field, control, or property. For example, Forms![Orders]![OrderID] is an identifier that refers to the value in the OrderID control on the Orders form.

Image Frame

A control that is used to display a picture on a form, report, or data access page.

Image OLE Object Data Type

In an Access project, a variable-length data type that can hold a maximum of $2^{31} - 1$ (2,147,483,647) bytes of binary data. It is used to store BLOBs (Binary Large Objects), such as pictures, documents, sounds, and compiled code.

Import

To copy data from a text file, spreadsheet, or database table into an Access table. You can use the imported data to create a new table, or you can append (add) it to an existing table that has a matching data structure.

Import Export Spec

A specification that stores the information Access needs to import or export a fixed-width or delimited text file.

Index

A feature that speeds up searching and sorting in a table based on key values and can enforce uniqueness on the rows in a table. The primary key of a table is automatically indexed. Some fields can't be indexed because of their data type.

Indexes Window

In an Access database, a window in which either you can view or edit a table's indexes or you can create multiple-field indexes.

Indirect Synchronization

A synchronization method used in a disconnected environment, such as when you travel with a portable computer.

In-Place Activation

Activation of an OLE object's OLE server from within a field or control. For example, you can play a waveform audio (.wav) file contained in a control by double-clicking the control.

Input Mask

A format that consists of literal display characters (such as parentheses, periods, and hyphens) and mask characters that specify where data is to be entered as well as what kind of data and how many characters are allowed.

Instance

An object created from the class that contains its definition. For example, multiple instances of a form class share the same code and are loaded with the same controls that were used to design the form class.

Int Data Type

In an Access project, a data type of 4 bytes (32 bits) that stores whole numbers in the range of -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647).

Integer Data Type

A fundamental data type that holds integers. An **Integer** variable is stored as a 16-bit (2-byte) number ranging in value from -32,768 to 32,767.

Internet Synchronization

Used to synchronize replicas in a disconnected environment in which an Internet server is configured.

Intrinsic Constant

A constant supplied by Access, VBA, ADO, or DAO. These constants are available in the Object Browser by clicking **globals** in each of these libraries.

Item

A unique element of data within a field. When a lower level of items is available for display in a PivotTable list or the field list, an expand indicator (+) appears beside the item.

J

[Back to top](#)

Jet Replication Objects (JRO)

A set of automation interfaces that you can use to perform actions specific to Microsoft Jet databases. Using JRO, you can compact databases, refresh data from the cache, and create and maintain replicated databases.

K

[Back to top](#)

Keyboard Handler

Code that determines and responds to keys or key combinations pressed by the user.

L

[Back to top](#)

Label

A control that displays descriptive text, such as a title, a caption, or

instructions, on a form, report or data access page. Labels might or might not be attached to another control.

Layout Preview

A preview of a report that uses a minimum amount of data to display the general layout of the report. Click Print Preview to display the exact data that appears in the report.

Left Outer Join

An outer join in which all the records from the left side of the LEFT JOIN operation in the query's SQL statement are added to the query's results, even if there are no matching values in the joined field from the table on the right.

Legend

A box that identifies the patterns or colors assigned to data series or categories in a chart.

Library Database

A collection of procedures and database objects that you can call from any application. In order to use the items in the library, you must first establish a reference from the current database to the library database.

Link

A connection between an OLE object and its OLE server, or between a dynamic data exchange (DDE) source document and a destination document.

List Index

The sequence of numbers for items in a list, starting with 0 for the first item, 1 for the second item, and so forth.

Local Object

A table, query, form, report, macro, or module that remains in the replica or Design Master where it was created. Neither the object nor changes to the object are copied to other members in the replica set.

Local Replica

A replica that exchanges data with its hub or a global replica, but not with other replicas in the replica set.

Locked

The condition of a record, recordset, or database that makes it read-only to all users except the user currently modifying it.

Lookup Field

A field, used on a form or report in an Access database, that either displays a list of values retrieved from a table or query, or stores a static set of values.

M

[Back to top](#)

Macro

An action or set of actions you can use to automate tasks.

Macro Group

A collection of related macros that are stored together under a single macro name. The collection is often referred to simply as a macro.

Main Form

A form that contains one or more subforms.

Make Table Query

A query (SQL statement) that creates a new table and then creates records (rows) in it by copying records from an existing table.

Manual Link

A link that requires you to take action to update your data after the data in the source document changes.

Many-To-Many Relationship

An association between two tables in which one record in either table can relate to many records in the other table. To establish one, create a third table and add the primary key fields from the other two tables to this table.

Marquee

Moving text used on a data access page to draw the user's attention to a specific page element, such as a headline or an important announcement. To place a marquee on a page, create a scrolling text control.

Maximum Record Limit

To improve performance, you can specify the maximum number of records that will be retrieved from a Microsoft SQL Server database for a form or datasheet in an Access project.

MDE File

Microsoft Access database (.mdb) file with all modules compiled and all editable source code removed.

Memo Data Type

Memo data type: In a Microsoft Access database, this is a field data type. Memo fields can contain up to 65,535 characters.

Microsoft Access Database

A collection of data and objects, such as tables, queries, or forms, related to a particular topic or purpose. The Microsoft Jet database engine manages the data.

Microsoft Access Object

An object defined by Access that relates to Access, its interface, or an application's forms, reports, and data access pages, and that you use to program the elements of the interface used for entering and displaying data.

Microsoft Data Engine (MSDE)

A client/server data engine that provides local data storage on a smaller computer system, such as a single-user computer or small workgroup server, and is compatible with Microsoft SQL Server 7.0.

Microsoft Jet Database Engine

The part of the Access database system that retrieves and stores data in user and system databases. It can be thought of as a data manager upon which database systems, such as Access, are built.

Microsoft SQL Server Database

A database in Microsoft SQL Server, it consists of tables, views, indexes, stored procedures, functions, and triggers.

Module Level

Describes any variable or constant declared in the Declarations section of a module or outside a procedure. Variables or constants declared at the module level are available to all procedures in a module.

Module Level Variable

A variable that's declared in the Declarations section of a module by using the **Private** keyword. These variables are available to all procedures in the module.

Money Data Type

In an Access project, a data type that stores monetary values in the range -922,337,203,685,477.5707 through 922,337,203,685,477.5807, with accuracy to a ten-thousandth of a monetary unit.

Move Handle

The large square displayed in the upper-left corner of the selected control in Design view. You can click and move the handle to move the control to another location.

Move Mode

The mode in which you can move column(s) in Datasheet view by using the left and right arrow keys.

MultiUser Database

A database that permits more than one user to access and modify the same set of data at the same time.

N

[Back to top](#)

Name Auto Correct

A feature that automatically corrects common side effects that occur when you rename forms, reports, tables, queries, or fields, or controls on forms and reports. However, Name AutoCorrect cannot repair all references to renamed objects.

Navigation Buttons

The buttons that you use to move through records. They are located in the lower-left corner of the Datasheet view and Form view windows. They are also available during previewing so that you can move through the pages of your document.

Nchar Data Type

In an Access project, a fixed-length data type with a maximum of 4,000 Unicode characters. Unicode characters use 2 bytes per character and support all international characters.

Normalize

To minimize the duplication of information in a relational database through effective table design. You can use the Table Analyzer Wizard to normalize your database.

Ntext Data Type

In an Access project, a variable-length data type that can hold a maximum of $2^{30} - 1$ (1,073,741,823) characters. Columns with the ntext data type store a 16-byte pointer in the data row, and the data is stored separately.

Null

A value you can enter in a field or use in expressions or queries to indicate missing or unknown data. In Visual Basic, the **Null** keyword indicates a **Null** value. Some fields, such as primary key fields, can't contain **Null**.

Null Field

A field containing a **Null** value. A null field isn't the same as a field that contains a zero-length string ("") or a field with a value of 0.

Number Data Type

Number data type: In a Microsoft Access database, a field data type designed for numerical data that will be used in mathematical calculations. Use the Currency data type, however, to display or calculate currency values.

Numeric Data Type

In an Access project, an exact numeric data type that holds values from

-10³⁸ - 1 through 10³⁸ - 1. You can specify the scale (maximum total number of digits) and precision (maximum number of digits to the right of the decimal point).

Nvarchar Data Type

In an Access project, a variable-length data type with a maximum of 4,000 Unicode characters. Unicode characters use 2 bytes per character and support all international characters.

O

[Back to top](#)

Object Application

An application or DLL that supplies a linked or embedded OLE object to another application. For example, if an OLE object in an Access database contains a Microsoft Excel worksheet, Excel is the OLE server.

Object Data Type

A fundamental data type representing any object that can be recognized by Visual Basic. Although you can declare any object variable as type **Object**, it's best to declare object variables according to their specific type.

Object Library

A file that contains definitions of objects and their methods and properties. The file that contains an object library typically has the extension .olb.

ODBC Connect String Builder

An Access tool you can use to connect to an SQL database when you create a pass-through query. If you save the query, the connection string is stored with the query.

ODBC Data Source

Data and the information needed to access that data from programs and databases that support the Open Database Connectivity (ODBC) protocol.

ODBC Database

A database for which an Open Database Connectivity (ODBC) driver that you can use for importing, linking to, or exporting data is supplied.

ODBC Direct

A technology that allows you to access ODBC data sources directly by using DAO features that bypass the Microsoft Jet database engine.

OLE Object

An object supporting the OLE protocol for object linking and embedding. An OLE object from an OLE server, for example, a Windows Paint picture

or a Microsoft Excel spreadsheet can be linked or embedded in a field, form, or report.

OLE Object Data Type

A field data type you use for objects created in other applications that can be linked or embedded (inserted) in an Access database.

OLEDB

A component database architecture that provides efficient network and Internet access to many types of data sources, including relational data, mail files, flat files, and spreadsheets.

OLEDB Provider

A program in the OLE DB architecture that enables native access to data, instead of accessing data using ODBC or IISAM drivers, which are external ways to access the data.

One-To-Many

An association between two tables in which the primary key value of each record in the primary table corresponds to the value in the matching field or fields of many records in the related table.

One-To-One

An association between two tables in which the primary key value of each record in the primary table corresponds to the value in the matching field or fields of one and only one record in the related table.

Option Button

A control, also called radio button, that is typically used as part of an option group to present alternatives on a form, report, or data access page. The user cannot select more than one option.

Option Group

A frame that can contain check boxes, toggle buttons, and option buttons on a form, report, or data access page. You use an option group to present alternatives from which the user can select a single option.

Outer Join

A join in which each matching record from two tables is combined into one record in the query's results, and one table contributes all of its records, even if the values in the joined field don't match those in the other table.

Owner

When security is being used, the user account that has control over a database or database object. By default, the user account that created a database or database object is the owner.

P

[Back to top](#)

Page

A portion of the database (.mdb) file in which record data is stored. Depending on the size of the records, a page (4 KB in size) may contain more than one record.

Page Footer

Use to display page summaries, dates, or page numbers at the bottom of every page in a form or report. In a form, the page footer appears only when you print the form.

Page Header

Use to display a title, column headings, dates, or page numbers at the top of every page in a form or report. In a form, the page header appears only when you print the form.

Page View

An Access window in which you can browse the contents of a data access page. Pages have the same functionality in Page view as in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

Parameter Query

A query in which a user interactively specifies one or more criteria values. A parameter query is not a separate kind of query; rather, it extends the flexibility of a query.

Partial Replica

A database that contains only a subset of the records in a full replica. With a partial replica, you can set filters and identify relationships that define which subset of the records in the full replica should be present in the database.

Pass Through Query

An SQL-specific query you use to send commands directly to an ODBC database server. By using pass-through queries, you work directly with the tables on the server instead of having the Microsoft Jet database engine process the data.

Permissions

A set of attributes that specify what kind of access a user has to data or objects in a database.

Pin

A case-sensitive alphanumeric string 4 to 20 characters long that Access uses in combination with the account name to identify a user or group in an

Access workgroup.

PivotChart View

A view that shows a graphical analysis of data in a datasheet or form. You can see different levels of detail or specify the layout by dragging fields and items, or by showing and hiding items in the drop-down lists for the fields.

PivotTable

An interactive table that summarizes large amounts of data by using format and calculation methods that you choose. You can rotate its row and column headings to view the data in different ways, similar to an Excel PivotTable report.

PivotTable List

A Microsoft Office Web Component used to analyze data interactively on a Web page. Data displayed in a row and column format can be moved, filtered, sorted, and calculated in ways that are meaningful for your audience.

PivotTable View

A view that summarizes and analyzes data in a datasheet or form. You can use different levels of detail or organize data by dragging the fields and items, or by showing and hiding items in the drop-down lists for the fields.

Plus Pointer

The pointer that appears when you move the pointer to the left edge of a field in a datasheet. When the plus pointer appears, you can click to select the entire field.

Pop-up

A form that stays on top of other windows. A pop-up form can be modal or modeless.

Primary Key

One or more fields (columns) whose values uniquely identify each record in a table. A primary key cannot allow **Null** values and must always have a unique index. A primary key is used to relate a table to foreign keys in other tables.

Primary Table

The "one" side of two related tables in a one-to-many relationship. A primary table should have a primary key and each record should be unique.

Private Procedure

A **Sub** or **Function** procedure is declared as private by using the **Private** keyword in a **Declare** statement. Private procedures are available for use only by other procedures within the same module.

Procedure

A sequence of declarations and statements in a module that are executed as a unit. Procedures in Visual Basic include both **Sub** and **Function** procedures.

Procedure Level

Describes any variables or constants declared within a procedure. Variables and constants declared within a procedure are available to that procedure only.

Procedure Level Variable

A variable that's declared within a procedure. Procedure-level variables are always private to the procedure in which they're declared.

Project

The set of all code modules in a database, including standard modules and class modules. By default, the project has the same name as the database.

Property Sheet

A window used to view or modify the properties of various objects such as tables, queries, fields, forms, reports, data access pages, and controls.

Publication

In an Access project, a publication can contain one or more published tables or stored procedure articles from one user database. Each user database can have one or more publications. An article is a grouping of data replicated as a unit.

Publish

The process of exporting datasheets, forms, or reports to static HTML or server-generated HTML, or creating data access pages, and setting up these files and all related files as a Web application on a Web server such as Microsoft IIS.

Q

[Back to top](#)

QBEGrid

The grid that you use to design a query or filter in query Design view or in the Advanced Filter/Sort window. For queries, this grid was formerly known as the QBE grid.

Query

A question about the data stored in your tables, or a request to perform an action on the data. A query can bring together data from multiple tables to serve as the source of data for a form, report, or data access page.

Query Window

A window in which you work with queries in Design view, Datasheet view, SQL view, or Print Preview.

R

[Back to top](#)

Real Data Type

In an Access project, an approximate numeric data type with seven-digit precision. It can hold positive values from approximately $1.18E - 38$ through $3.40E + 38$, negative values from approximately $-1.18E - 38$ through $-3.40E + 38$, or zero.

Record Navigation Control

A control used on a data access page to display a record navigation toolbar. In a grouped page, you can add a navigation toolbar to each group level. You can customize the record navigation control by changing its properties.

Record Navigation Section

A section in a data access page that displays the record navigation control for a specific group level. It appears after the corresponding group header section. You can't place bound controls in a record navigation section.

Record Number Box

A small box that displays the current record number in the lower-left corner in Datasheet view and Form view. To move to a specific record, you can type the record number in the box, and press ENTER.

Record Selector

A small box or bar to the left of a record that you can click to select the entire record in Datasheet view and Form view.

Record Source

The underlying source of data for a form, report, or data access page. In an Access database, it could be a table, query, or SQL statement. In an Access project, it could be a table, view, SQL statement, or stored procedure.

Recordset

The collective name given to table-, dynaset-, and snapshot-type **Recordset** objects, which are sets of records that behave as objects.

Referenced Database

The Access database to which the user has established a reference from the current database. The user can create a reference to a database and then call procedures within standard modules in that database.

Referencing Database

The current Access database from which the user has created a reference to another Access database. The user can create a reference to a database and then call procedures within standard modules in that database.

Referential Integrity

Rules that you follow to preserve the defined relationships between tables when you enter or delete records.

Refresh

In an Access database, to redisplay the records in a form or datasheet to reflect changes that other users have made. In an Access project, to rerun a query underlying the active form or datasheet in order to reflect changes to records.

Relationship

An association established between common fields (columns) in two tables. A relationship can be one-to-one, one-to-many, or many-to-many.

Relationships Window

A window in which you view, create, and modify relationships between tables and queries.

Relative Positioning

Places the element in the natural HTML flow of the document but offsets the position of the element based on the preceding content.

Repaint

To redraw the screen. The **Repaint** method completes any pending screen updates for a specified form.

Replica

A copy of a database that is a member of a replica set and can be synchronized with other replicas in the set. Changes to the data in a replicated table in one replica are sent and applied to the other replicas.

Replica Set

The Design Master and all replicas that share the same database design and unique replica set identifier.

Replica Set Topology

The order in which changes are propagated from replica to replica. Topology determines how quickly changes in another replica appear in your replica.

Replication

The process of copying a database so that two or more copies can exchange updates of data or replicated objects. This exchange is called synchronization.

Report

An Access database object that prints information formatted and organized according to your specifications. Examples of reports are sales summaries, phone lists, and mailing labels.

Report Footer

A report section used to place information that normally appears at the bottom of the page, such as page numbers, dates, and sums.

Report Header

A report section used to place information, such as a title, date, or report introduction, at the beginning of a report.

Report Module

A module that includes code for all event procedures triggered by events occurring on a specific report or its controls.

Report Selector

The box where the rulers meet in the upper-left corner of a report in Design view. Use the box to perform report-level operations, such as selecting the report.

Report Snapshot

A file (.snp extension) that contains a high-fidelity copy of each page of an Access report. It preserves the two-dimensional layout, graphics, and other embedded objects of the report.

Report Window

A window in which you work with reports in Design view, Layout Preview, or Print Preview.

Requery

To rerun a query underlying the active form or datasheet in order to reflect changes to the records, display newly added records, and eliminate deleted records.

Reserved Word

A word that's part of the Visual Basic for Applications programming language. Keywords include the names of statements, data types, methods, operators, objects, and predefined functions. Don't use keywords for variable or object names.

Right Outer Join

An outer join in which all the records from the right side of the RIGHT JOIN operation in the query's SQL statement are added to the query's results, even if there are no matching values in the joined field from the table on the left.

Row Area

The part of PivotTable view that contains row fields.

Row Field

A field in the row area of PivotTable view. Items in row fields are listed down the left side of the view. Inner row fields are closest to the detail area; outer row fields are to the left of the inner row fields.

Row Selector

A small box or bar that, when clicked, selects an entire row in table or macro Design view, or when you sort and group records in report Design view.

S

[Back to top](#)

Script Window

The window in which you create and modify macros.

Scrolling Text Control

A control used on a data access page to display a marquee, or moving text. You can customize the display of text by setting options such as direction and speed. Browsers that do not support scrolling text will display static text.

Section

A part of a form, report, or data access page such as a header, footer, or detail section.

Section Bar

The horizontal bar above a data access page section in Design view. The section bar displays the type and name of the section. Use it to access the section's property sheet.

Section Selector

The box on the left side of a section bar when an object is open in Design view. Use the box to perform section-level operations, such as selecting the section.

Secure System

An Access workgroup in which users log on with a user name and password and access to database objects is restricted according to permissions granted to specific user accounts and groups.

Select Query

A query that asks a question about the data stored in your tables and returns a result set in the form of a datasheet without changing the data.

Selection Rectangle

The rectangle formed by the currently selected rows (records) and columns (fields) within Datasheet view.

Self Join

A join in which a table is joined to itself. Records from the table are combined with other records from the same table when there are matching values in the joined fields.

Separator

A character that separates units of text or numbers.

Series Field

A field displayed in the series area of a chart and containing series items. A series is a group of related data points.

Series Point

An individual data value plotted in a chart and represented by a column, bar, line, pie or doughnut slice, or other type of data marker.

Server Filter By Form

A technique that uses a version of the current form or datasheet with empty fields in which you can type values you want the filtered records to contain. The data is filtered by the server before it's retrieved from the database.

Server-generated HTML

An Active Server Pages (ASP) or IDC/HTX file that is output from a table, query, or form, connected to an ODBC data source, and processed by the Internet Information Server to dynamically create read-only HTML files.

Session

A sequence of operations performed by the Microsoft Jet database engine that begins when a user logs on and ends when the user logs off. All operations during a session form one transaction scope and are subject to the user's logon permissions.

Smalldatetime Data Type

In an Access project, a date and time data type that is less precise than the datetime data type. Data values range from January 1, 1900, through June 6, 2079, to an accuracy of one minute.

Smallint Data Type

In an Access project, a data type of 2 bytes (16 bits) that stores whole numbers in the range of -2^{15} (-32,768) through $2^{15} - 1$ (32,767).

Smallmoney Data Type

In an Access project, a data type that stores monetary values from -214,748.3648 to 214,748.3647, with accuracy to a ten-thousandth of a monetary unit. When smallmoney values are displayed, they are rounded up

to two decimal places.

Snapshot

A static image of a set of data, such as the records displayed as the result of a query. Snapshot-type **Recordset** objects can be created from a base table, a query, or another recordset.

Snapshot Viewer

A program that you can use to view, print, or mail a snapshot, such as a report snapshot. Snapshot Viewer 11.0 consists of a stand-alone executable program, a Snapshot Viewer control (Snapview.ocx), and other related files.

Snapshot Viewer Control

An ActiveX control (Snapview.ocx) that you use to view a snapshot report from Microsoft Internet Explorer 3.0 or later, or from any application that supports ActiveX controls, such as Access or Microsoft Visual Basic.

SQL Database ODBC

A database based on Structured Query Language (SQL).

SQL Specific Query

A query that consists of an SQL statement. Subqueries and pass-through, union, and data-definition queries are SQL-specific queries.

SQL String

An expression that defines an SQL command, such as SELECT, UPDATE, or DELETE, and includes clauses such as WHERE and ORDER BY. SQL strings/statements are typically used in queries and in aggregate functions.

SQL Variant Data Type

In an Access project, a data type that stores values of several data types, except for text, ntext, image, timestamp, and sql_variant types. It is used in a column, parameter, variable, or return value of a user-defined function.

SQL View

A window that displays the SQL statement for the current query or is used to create an SQL-specific query (union, pass-through, or data definition). When you create a query in Design view, Access constructs the SQL equivalent in SQL view.

Standard Module

A module in which you can place **Sub** and **Function** procedures that you want to be available to other procedures throughout your database.

Stored Procedure

A precompiled collection of SQL statements and optional control-of-flow statements stored under a name and processed as a unit. They are stored in an SQL database and can be run with one call from an application.

String Delimiter

Text characters that set apart a string embedded within a string. Single quotation marks (') and double quotation marks (") are string delimiters.

Sub Procedure

A procedure that carries out an operation. Unlike a **Function** procedure, a **Sub** procedure doesn't return a value. You begin a **Sub** procedure with a **Sub** statement and end it with an **End Sub** statement.

Subdatasheet

A datasheet that is nested within another datasheet and that contains data related or joined to the first datasheet.

Subform

A form contained within another form or a report.

Subform Subreport

A control that displays a subform in a form or a subform or a subreport in a report.

Subquery

An SQL SELECT statement inside another select or action query.

Subreport

A report contained within another report.

Subscribe

To agree to receive a publication in an Access database (.mdb) or an Access project (.adp). A subscriber database subscribes to replicated data from a publisher database.

Subscription

The database that receives tables and data replicated from a publisher database in an Access project (.adp).

Synchronization

The process of updating two members of a replica set by exchanging all updated records and objects in each member. Two replica set members are synchronized when the changes in each have been applied to the other.

Sysname Data Type

In an Access project, a special system-supplied, user-defined data type that is used for table columns, variables, and stored procedure parameters that store object names.

System Objects

Database objects that are defined by the system, such as the table MSysIndexes, or by the user. You can create a system object by naming the object with USys as the first four characters in the object name.

T

[Back to top](#)

Tab Control

A control you can use to construct a single form or dialog box that contains several pages, each with a tab, and each containing similar controls, such as text boxes or option buttons. When a user clicks a tab, that page becomes active.

Table

A database object that stores data in records (rows) and fields (columns). The data is usually about a particular category of things, such as employees or orders.

Table Data Type

In an Access project, a special data type used to store a result set in a local variable or return value of a user-defined function for later processing. It can be used in place of a temporary table stored in the tempdb database.

Table Properties

In an Access database, attributes of a table that affect the appearance or behavior of the table as a whole. Table properties are set in table Design view, as are field properties.

Table Window

In an Access database, a window in which you work with tables in Design view or Datasheet view.

Text Box

A control, also called an edit field, used on a form, report, or data access page to display text or accept data entry. It can have a label attached to it.

Text Data Type

In a Microsoft Access database, this is a field data type. Text fields can contain up to 255 characters or the number of characters specified by the **Fieldsize** property, whichever is less.

Textmemo Data Type

In an Access project, a variable-length data type that can hold a maximum of $2^{31} - 1$ (2,147,483,647) characters; default length is 16.

Theme

A collection of formats that determines the appearance of the controls, sections, body, and text on a data access page.

Timestamp Data Type

In an Access project, a data type that is automatically updated every time a row is inserted or updated. Values in timestamp columns are not datetime

data, but binary(8) or varbinary(8), indicating the sequence of data modifications.

Tinyint Data Type

In an Access project, a data type of 1 byte (8 bits) that stores whole numbers in the range of 0 through 255.

Toggle Button

A control used to provide on/off options on a form or report. It can display either text or a picture, and can be standalone or part of an option group.

Tool Properties

The default property setting of a control type. You customize a control type before you create two or more similar controls to avoid customizing each control individually.

Toolbox

A set of tools that are available in Design view to add controls to a form, report, or data access page. The toolset available in page Design view is different from the toolset available in form and report Design view.

ToolTips

Brief descriptions of the names of buttons and boxes on toolbars and in the toolbox. A ToolTip is displayed when the mouse pointer rests on the button or combo box.

Topology

The order in which changes are propagated from replica to replica. Topology is important because it determines how quickly changes in another replica appear in your replica.

Total Field

A field that summarizes data from the underlying record source. A total field might use a summary function, such as Sum or Count, or use an expression to calculate summary values.

Totals Query

A query that displays a summary calculation, such as an average or sum, for values in various fields from a table or tables. A totals query is not a separate kind of query; rather, it extends the flexibility of select queries.

Transaction

A series of changes made to a database's data and schema.

Trigger

A special form of a stored procedure that is carried out automatically when data in a specified table is modified. Triggers are often created to enforce referential integrity or consistency among logically related data in different tables.

U

[Back to top](#)

Unbound Field

A control that is not connected to a field in an underlying table, query, or SQL statement. An unbound control is often used to display informational text or decorative pictures.

Unbound Form Report

A form or report that isn't connected to a record source such as a table, query, or SQL statement. (The form's or report's **RecordSource** property is blank.)

Unbound Object Frame

A control you place on a form or report to contain an unbound object. An unbound object is an object, such as a picture, whose value isn't derived from data stored in a table.

Union Query

A query that uses the UNION operator to combine the results of two or more select queries.

Unique Index

An index defined by setting a field's **Indexed** property to **Yes (No Duplicates)**. A unique index will not allow duplicate entries in the indexed field. Setting a field as the primary key automatically defines the field as unique.

Uniqueidentifier Data Type

In an Access project, a 16-byte globally unique identifier (GUID).

Updatable Snapshot

A type of recordset that works efficiently in a client/server environment by caching data on the client and minimizing round trips to the server to access and update data.

Update

To accept changes to data in a record. The changes are saved in the database when you move to another record on a form, datasheet, or data access page, or when you explicitly save the record.

Update Query

An action query (SQL statement) that changes a set of records according to criteria (search conditions) that you specify.

User Account

An account identified by a user name and personal ID (PID) that is created

to manage the user's permissions to access database objects in an Access workgroup.

User-Defined Collection

A collection that you create by adding objects to a **Collection** object. Items in a collection defined by the **Collection** object are indexed beginning with 1.

User-Defined Data Type

In a Microsoft SQL Server database, a definition of the type of data a column can contain. It is defined by the user with existing system data types. Rules and defaults can only be bound to user-defined data types.

User-Defined Function

A query that takes input parameters and returns a result like a stored procedure. Types: scalar (multistatement; returns one value), inline (one statement; an updateable table value), and table (multistatement; table value).

User-Defined Object

A custom object defined in a form or report class module. In a class module, you can create methods and properties for a new object, create a new instance of this object, and manipulate it with these properties and methods.

User-Level Security

When using user-level security in an Access database, a database administrator or an object's owner can grant individual users or groups of users specific permissions to tables, queries, forms, reports, and macros.

Users Group

The group account that contains all user accounts. Access automatically adds user accounts to the Users group when you create them.

V

[Back to top](#)

Validation

The process of checking whether entered data meets certain conditions or limitations.

Validation Rule

A property that defines valid input values for a field or record in a table, or a control on a form. Access displays the message specified in the **ValidationText** property when the rule is violated.

Varbinary Data Type

In an Access project, a variable-length data type with a maximum of 8,000 bytes of binary data.

Varchar Data Type

In an Access project, a variable-length data type with a maximum of 8,000 ANSI characters.

View object

In an Access project, a type of query that is a virtual table based on an SQL SELECT statement. For example, a view might only contain 3 out of 10 available columns in a join of two tables, in order to limit access to certain data.

Visibility

A property of a replica that indicates which members of the replica set it can synchronize with and which conflict resolution rules apply. Replicas fall into three visibility types: global, local, and anonymous.

Visual Basic Identifier

A data member in a Visual Basic code module. An identifier can be a **Sub**, **Function**, or **Property** procedure, a variable, a constant, a **Declare** statement, or a user-defined data type.

W

[Back to top](#)

Web Page Preview

Opens a data access page for preview in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

Where Clause

The part of an SQL statement that specifies which records to retrieve.

WorkGroup

A group of users in a multiuser environment who share data and the same workgroup information file.

Workgroup ID

A case-sensitive alphanumeric string 4 to 20 characters long that you enter when creating a new workgroup information file by using the Workgroup Administrator. This uniquely identifies the Admins group for this workgroup file.

Workgroup Information File

A file Access reads at startup that contains information about the users in a

workgroup. This information includes users' account names, their passwords, and the groups of which they are members.

X

[Back to top](#)

XML Attribute

Information added to a tag to provide more information about the tag, such as `<ingredient quantity="2" units="cups">flour</ingredient>`.

XML Element

Information delimited by a start and end tag in an Extended Markup Language (XML) document. An example would be `<LastName>Davolio</LastName>`.

XML Entities

Combinations of characters and symbols that replace other characters when an XML document is parsed, usually those that have other meanings in XML. For example, `<` represents the `<` symbol which is also the opening bracket for a tag.

Y

[Back to top](#)

Yes/No DataType

Yes/No data type: A field data type you use for fields that will contain only one of two values, such as Yes or No and True or False. Null values are not allowed.

Z

[Back to top](#)

Zero-Length String

A string that contains no characters. You can use a zero-length string to indicate that you know there's no value for a field. You enter a zero-length string by typing two double quotation marks with no space between them (`""`).

[Show All](#)

Access specifications

Microsoft Access database specifications

[Access database](#)

Attribute	Maximum
Microsoft Access database (.mdb) file size	2 gigabytes minus the space needed for system objects .
Number of objects in a database	32,768
Modules (including forms and reports with the HasModule property set to True)	1,000
Number of characters in an object name	64
Number of characters in a password	14
Number of characters in a user name or group name	20
Number of concurrent users	255

[Table](#)

Attribute	Maximum
Number of characters in a table name	64
Number of characters in a field name	64
Number of fields in a table	255
Number of open tables	2048; the actual number may be less because of tables opened internally by Microsoft Access
Table size	2 gigabyte minus the space needed for the system objects
Number of characters in a Text field	255

Number of characters in a Memo field	65,535 when entering data through the user interface; 1 gigabyte of character storage when entering data programmatically
Size of an OLE Object field	1 gigabyte
Number of indexes in a table	32
Number of fields in an index	10
Number of characters in a validation message	255
Number of characters in a validation rule	2,048
Number of characters in a table or field description	255
Number of characters in a record (excluding Memo and OLE Object fields)	2,000
Number of characters in a field property setting	255

Query

Attribute	Maximum
Number of enforced relationships	32 per table minus the number of indexes that are on the table for fields or combinations of fields that are not involved in relationships
Number of tables in a query	32
Number of fields in a recordset	255
Recordset size	1 gigabyte
Sort limit	255 characters in one or more fields
Number of levels of nested queries	50
Number of characters in a cell in the query design grid	1,024

Number of characters for a parameter in a parameter query 255

Number of ANDs in a WHERE or HAVING clause 99

Number of characters in an SQL statement approximately 64,000

[Form and report](#)

Attribute	Maximum
Number of characters in a label	2,048
Number of characters in a text box	65,535
Form or report width	22 in. (55.87 cm)
Section height	22 in. (55.87 cm)
Height of all sections plus section headers (in Design view)	200 in. (508 cm)
Number of levels of nested forms or reports	7
Number of fields or expressions you can sort or group on in a report	10
Number of headers and footers in a report	1 report header/footer; 1 page header/footer; 10 group headers/footers
Number of printed pages in a report	65,536
Number of controls and sections you can add over the lifetime of the form or report	754
Number of characters in an SQL statement that serves as the Recordsource or Rowsource property of a form, report, or control (both .mdb and .adp)	32,750

[Macro](#)

Attribute	Maximum
Number of actions in a macro	999

Number of characters in a condition	255
Number of characters in a comment	255
Number of characters in an action argument	255

Microsoft Access project specifications

[Access project](#)

Attribute	Maximum
Number of objects in a Microsoft Access project (.adp)	32,768
Modules (including forms and reports with the HasModule property set to True)	1,000
Number of characters in an object name	64
	250 (Microsoft SQL Server 6.5)
Number of columns in a table	1024 (Microsoft SQL Server 7.0 and 2000)

[Microsoft SQL Server database](#)

Microsoft SQL Server maximum capacity specifications are described in the SQL Server documentation. For more information about SQL Server Books Online, see the [Microsoft Developer Network](#) Web site

[Form and report](#)

Attribute	Maximum
Number of characters in a label	2,048
Number of characters in a text box	65,535
Form or report width	22 in. (55.87 cm)
Section height	22 in. (55.87 cm)
Height of all sections plus section headers (in Design view)	200 in. (508 cm)
Number of levels of nested forms or reports	7
Number of fields or expressions you can sort or group on in a report	10

Number of headers and footers in a report	1 report header/footer; 1 page header/footer; 10 group headers/footers
Number of printed pages in a report	65,536
Number of controls and sections you can add over the lifetime of the form or report	754
Number of characters in an SQL statement that serves as the Recordsource or Rowsource property of a form, report, or control (both .mdb and .adp)	32,750

Macro

Attribute	Maximum
Number of actions in a macro	999
Number of characters in a condition	255
Number of characters in a comment	255
Number of characters in an action argument	255

[Show All](#)

Ways to get help on writing code

Some of the content in this topic may not be applicable to some languages.

There are many ways to access Help for answers to questions about [Microsoft Visual Basic](#) and for information about Visual Basic keywords. To access Help, you can:

- **Press F1 in a Visual Basic module** While you are working in a [module](#), you can go directly to Help on a method, property, function, statement, or object by clicking the keyword in a line of code and then pressing F1.
- **Press F1 in the Object Browser** You can use the [Object Browser](#) to see information about objects, properties, methods, and constants.
- **Ask the Microsoft Office Assistant** To display the assistant balloon and type an entry, either click Office Assistant on the [toolbar](#), or if the Assistant is already displayed with no balloon, click the Assistant. The Assistant makes it easy to find the topics you're looking for. Just type the method, property, statement, or function you want help on, or type a question, and then select one of the topics the Assistant suggests. The more specific your question is, the more specific the suggested topics will be. An example of a specific entry is "Program my form to automatically update data."
- **Use the Contents tabs** Click the Contents tab, and then double-click the Programming Information book to see the categories of reference information, such as actions, events, functions, objects and collections, methods, properties, and statements. Double-click a category book to view lists of topics for each category. To print a topic or book of topics, select the topic or book and then click **Print**.
- **Switch between your application and language reference Help** Press ALT+TAB or use the Microsoft Windows taskbar. If you leave Help active, you can immediately return to the last topic you were viewing.

[Show All](#)

Get help on Access events and properties

You can get help on a specific event or property, such as the **BeforeUpdate** event, the **Format** property, or the **OnClick** event, in one of the three ways:

[From the property sheet](#)

1. In Design view, select the object whose property or event you want help on.
2. Press F4 to display the object's property sheet.
3. On the **All** tab, click the property or event.
4. Press F1.

[From the Visual Basic Editor](#)

1. In the **Database** window, press ALT+F11 to start the Visual Basic Editor.
2. On the **Insert** menu, click **Module**.
3. Type the name of the event or property on which you want to get help.
4. Place the cursor inside the string, or right after the last character of the string.
5. Press F1.

Notes

- You can also press F1 in the **Object Browser** to get help on the selected Visual Basic keyword.
- Press ALT+TAB to switch between your application and the Visual Basic Editor.

[From the Table of Contents](#)

1. Click **Microsoft Access Help** on the toolbar.
2. On the **Microsoft Access Help** pane, click **Table of Contents**.
3. Under **Microsoft Access Visual Basic Reference**, look for the event or property.

4. Click the title to display the corresponding help topic.

Note Typing a Visual Basic keyword in the **Search** box on the **Microsoft Access Help** pane will not return help on the keyword.

[Show All](#)

About getting Access product help while in the Visual Basic Editor

The Help topics for Microsoft Access are divided into topics about Access features and programming-related topics. While there isn't a clear distinction as to what constitutes Access features versus programming-related features (some features appear in both such as properties that appear in Access property sheets), generally Access features are those features that don't require the user to write [Microsoft Visual Basic for Applications](#) code or access the [ADO](#) object model. Programming-related features include methods, events, controls, and anything specific to the ADO object model. Programming-related features are usually accessed in the [Visual Basic Editor](#) or in the [property sheet](#) of an [object](#). Pressing F1 when the cursor is in a field in a property sheet will always bring up Help.

When you're working with Access features, the Help system queries the Access feature topics; when you're in the Visual Basic Editor, the Help system queries the programming topics.

This result is that Help topics on programming features are not available when you're working with Access features, and vice versa. This organization simplifies the Help system and keeps the query results to a minimum number of relevant topics.

There are cases in which you might want to query both sets of Help topics. For example, if you are writing Visual Basic code to programmatically populate a report and you want to know more about adding a background picture or watermark to the report. Specific help for adding a background picture is only available in Help topics for Access features.

You can change the default setting so that the Help system searches both sets of topics by selecting the **Search for both product and programming help when programming** check box in the **Office Assistant** dialog box. Note that this option applies only when you are in the Visual Basic Editor.

Selecting this option can result in duplicate topics being displayed in the Help window. In most cases, the help you need on programming features will be

available while working with Access features, without your having to start the Visual Basic Editor.

Note You can start the Visual Basic Editor while in the [Database window](#) by pressing ALT+F11 or by going to the **Tools** menu, pointing to **Macro**, and then selecting **Visual Basic Editor**.

[Show All](#)

About getting help on SQL Server (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Microsoft Access documents in its Help system how to use and program a [Microsoft Access project](#) that is connected to a Microsoft SQL Server database. However, you may also need additional information about SQL Server to answer specific product questions you might have, such as reference information on Transact-SQL or the SQL Server Enterprise Manager.

 [Access to SQL Server Books Online and Transact SQL Reference information](#)

The primary source of information about Microsoft SQL Server is SQL Server Books Online. If you have access to SQL Server 2000 on either a local or remote computer, then you have access to SQL Server Books Online through the **Start** menu. However, if you don't have access to SQL Server Books Online, you can download it from the SQL Server Books Online download Web page.

 [An overview of the SQL Server documentation](#)

SQL Server documentation contains the following information.

Item	Description
SQL Server Books Online	Contains comprehensive documentation about SQL Server 2000 including: <ul style="list-style-type: none">• Conceptual information.• General instructions.• Reference information about Transact-SQL syntax.• Reference information about SQL Server error messages, numbers, and recovery actions.• General programming references.

Enterprise Manager Help	Help information about administering an SQL Server database.
Replication Wizard	Help information about configuring and managing replication.
Client Configuration Utility Help	Help information about managing SQL Server network connectivity options.
ODBC Driver Help	Help information about creating SQL Server ODBC data source definitions.
Profiler Help	Help information about capturing SQL Server events in a trace file for analysis and replay.
Query Analyzer Help	Help information about a graphical user interface that is used to interactively design and test Transact-SQL statements, batches, and scripts.
Data Transformation Services (DTS) Designer Help	Help information about importing, exporting, and transforming a variety of data sources.
SQL Distributed Management Objects (SQL-DMO)	Context-sensitive help information that is available from the development environment for SQL-DMO objects, properties and methods.
The Pubs Sample Database	The traditional SQL Server sample database.
The Northwind SQL Sample Database	A sample database similar to the Access Northwind sample database with specific SQL Server examples.

[Show All](#)

About getting Help on Microsoft Jet SQL

You can access the Microsoft Jet SQL Reference help topics while working in Access or from the Visual Basic Editor.

[From Access](#)

- **By searching for a Jet SQL keyword** In the **Access Help** pane, simply type a Jet SQL keyword, such as **WHERE** or **Drop** in the **Search** text box on the pane, and you will see a set of Jet SQL Help topics.

Note These topics do not include links to DAO reference topics or examples.

- **From the Table of Contents** Click **Table of Contents** on the **Access Help** pane. Under the **Creating and Working with Databases and Objects** folder, click **Database objects**, then click **Queries** to display the **Microsoft Jet SQL Reference** folder.

Note These topics do not include links to DAO reference topics or examples. If you want to display topics with links, open the Microsoft Jet SQL Reference folder located towards the bottom of the Table of Contents.

- **By pressing F1 from SQL View** Open a query in SQL view, and place the cursor on the keyword on which you want to display Help, and press F1.

[From the Visual Basic Editor](#)

- **From the Table of Contents** In the **Microsoft Visual Basic Help** pane, under **Table of Contents**, locate the **Microsoft Jet SQL Reference** folder. Open this folder and browse to the topic you want.

[Show All](#)

Get Script Editor help in a data access page

1. Open a [data access page](#) in [Design view](#) or [Page view](#).
2. On the **Tools** menu, point to **Macro**, and then click **Microsoft Script Editor**.
3. In the [Microsoft Script Editor](#), click **Help**  on the **Standard** toolbar, and then do one of the following:
 - To search for any words you want to find in the Script Editor Help, click **Answer Wizard**.
 - For a table of contents of Script Editor Help, click **Contents**.
 - To browse through an index of keywords to find information in the Script Editor Help, click **Index**.

Note The debugging feature of the Microsoft Script Editor does not work with Microsoft Windows Terminal Services.

[Show All](#)

Sample databases included with Access

Microsoft Access provides sample databases that you can use while you're learning Access.

 [Northwind Traders sample database](#)

The Northwind database and Access project (available from the **Sample Databases** command on the **Help** menu) contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world. By viewing the [database objects](#) included in the Northwind database, you can develop ideas for your own database application. You can also use the Northwind data to experiment with Access before you enter your own data. For example, you may want to practice designing queries using the Orders table in Northwind, since it contains enough records to produce meaningful results.

 [About creating the NorthwindCS.adp SQL Server database](#)

Unlike the sample [Microsoft Access database](#) (Northwind.mdb), the data and data definitions for the sample [Access project](#) (NorthwindCS.adp) need to be created the first time you open the project from the **Sample Databases** command on the **Help** menu. However, Access can automatically create the Northwind [SQL Server database](#) for you the first time you open the Northwind Access project.

The Startup form first asks you to log on to SQL Server, looks for Microsoft SQL Server 2000 Personal Desktop Engine or SQL Server running on your computer, and then does one of the following:

- If the Startup form finds SQL Server running on your computer, it asks if you want to create the NorthwindCS SQL Server database so that you can connect to it.

- If the Startup form doesn't find SQL Server running on your computer, it prompts you to enter a server name, user name, and password so you can connect to a SQL Server database on the network.

In either case if you say yes, the Northwind sample Access project runs the SQL installation script (NorthwindCS.SQL), located in the \Program Files\Microsoft Office\Office 11\Samples folder. This SQL script is a sequence of SQL statements that creates the SQL Server database data structure, extended properties, queries, and database diagrams, and then loads the sample data. Once the script has finished running, the Northwind sample Access project connects to the newly-created SQL server database.

 [Database wizard](#)

Microsoft Access also includes a Database Wizard that you can use to create common types of Access databases. You can use the databases exactly as created by the Database Wizard or as learning tools to help you design your own databases.

[Show All](#)

About the sample XSL transforms that come with Access

[Extensible Stylesheet Language \(XSL\)](#) style sheets are used to display [Extensible Markup Language \(XML\)](#) documents in a browser such as Microsoft Internet Explorer. The XSL style sheet is linked to the XML document and tells the browser how to display the data, allowing you to open the document directly in the browser. By using an XSL style sheet, you can select just the data you want to display, present that data in any order or grouping, and add or modify the information. You can also sort or filter the XML data, include [scripts](#) in the style sheet, and use programming methods to work with the data.

Microsoft Access 2003 comes with a sample XSL transform (XSLT) file, Rpt2HTM4.xsl, to assist you in creating your own XSL style sheet. This transform is a sample and should be tested thoroughly before use in your own application. This sample is used to transform ReportML (XML that describes an object's presentation-specific attributes) into a style sheet that will convert the XML data that Access exports into an HTML 4.0 data format document.

Access also comes with a featured transform titled Rpt2DAP.xsl. This file has been tested and is ready to use in your application. This file is used to transform ReportML into a data access page. Both files are located in "C:\Program Files\Microsoft Office\Office11\AccessWeb".

[Show All](#)

About accessibility for people with disabilities

[Accessibility features in Microsoft Access](#)

You can do the following to take advantage of the [accessibility](#) features in Access:

- View and print lists of all available [shortcut keys](#).
- Use the keyboard to define [relationships](#), add or copy text boxes or other [controls](#) on forms and reports, and rearrange columns in [Datasheet view](#).
- Create a [toolbar](#) that contains only the buttons and menus you use most often for the current database or for all your databases. You can also increase the size of the buttons and group related buttons together.
- Set menus to display all commands all the time. You can also change toolbars to appear as they did in earlier versions of Office.
- Create custom colors or change the background color for a [section](#) or control on a form or report.
- Insert repeated entries in a column quickly by using the keyboard.
- Automatically correct common typing errors and even specify a list of errors you want corrected as you type.

[Customizing the operating system](#)

Using the [operating system](#) running on your computer, you can set or change system [accessibility](#) options.

For example, you can use the Windows MouseKeys feature to set up the numeric keypad to move the mouse pointer and to click, double-click, and drag objects.

To access or change the accessibility options for any Windows operating system, press CTRL+ESC to open the Windows **Start** menu, choose **Settings**, choose **Control Panel**, and then choose the **Accessibility Options** icon. For Help on an option, select the option and press SHIFT+F1.

[Printed information about Microsoft services](#)

More information about Microsoft services for people with disabilities is available in an appendix in the book *Discovering the Microsoft Office 2003 Editions* which is included in the Office 2003 CD-ROM. For example, you'll find information about how people who are deaf or hard-of-hearing can contact the Microsoft Sales Information Center or the Microsoft Support Network. You'll also find information about obtaining Microsoft documentation for those who have difficulty reading or handling printed documentation. The appendix also describes third-party hardware and software products that make personal computers easier to use for people with disabilities, and lists organizations to contact for more information.

[Information on the Web](#)

If you have access to the Web, you can learn more about creating accessible content for Web pages by referring to the accessibility standards developed by the [World Wide Web Consortium](#). To learn more about accessibility in Microsoft products, visit the [Microsoft Accessibility](#) Web site.

[Show All](#)

Using accessibility features in Access

Microsoft Access contains [accessibility](#) features that make it easier to use for people with disabilities.

Use the keyboard instead of the mouse

[Use keyboard shortcuts](#)

You can use shortcut keys for quick access to frequently used commands or operations. The topics below list the shortcut keys available in Microsoft Access. You can also use access keys to move the focus to a menu, command, or control without using the mouse.

General shortcut keys

[Global Access shortcut keys](#)

[Opening databases](#)

CTRL+N

To open a new database

CTRL+O

To open an existing database

ALT+F4

To quit Microsoft Access

[Printing and saving](#)

CTRL+P

To print the current or selected object

P or CTRL+P

To open the **Print** dialog box

S

To open the **Page Setup** dialog box

C or ESC

To cancel Print Preview or Layout Preview

CTRL+S or SHIFT+F12 or ALT+SHIFT+F2

To save a database object

F12 or ALT+F2

To open the **Save As** dialog box

[Using a combo box or list box](#)

F4 or ALT+DOWN ARROW

To open a combo box

F9

To refresh the contents of a [Lookup field](#) list box or combo box

DOWN ARROW

To move down one line

PAGE DOWN

To move down one page

UP ARROW

To move up one line

PAGE UP

To move up one page

TAB

To exit the combo box or list box

[Finding and replacing text or data](#)

CTRL+F

To open the **Find** tab in the **Find and Replace** dialog box (Datasheet view and Form view only)

CTRL+H

To open the **Replace** tab in the **Find and Replace** dialog box (Datasheet view and Form view only)

SHIFT+F4

To find the next occurrence of the text specified in the **Find and Replace** dialog box when the dialog box is closed (Datasheet view and Form view only)

[Working in Design view](#)

F2

To switch between Edit mode (with insertion point displayed) and [Navigation mode](#)

F4

To switch to the property sheet (Design view in forms and reports in databases and Access projects)

F5

To switch to Form view from form Design view

F6

To switch between the upper and lower portions of a window (Design view of tables, macros, and queries and the Advanced Filter/Sort window only)

F7

To switch to the Code Builder from form or report Design view (Design view window or property sheet)

SHIFT+F7

To switch from the Visual Basic Editor to form or report Design view

ALT+V+P

To open property sheet for a selected object



[Editing controls in form and report Design view](#)

SHIFT+ENTER

To add a control to a section

CTRL+C

To copy the selected control to the Clipboard

CTRL+X

To cut the selected control and copy it to the Clipboard

CTRL+V

To paste the contents of the Clipboard in the upper-left corner of the selected section

CTRL+RIGHT ARROW

To move the selected control to the right

CTRL+LEFT ARROW

To move the selected control to the left

CTRL+UP ARROW

To move the selected control up

CTRL+DOWN ARROW

To move the selected control down

SHIFT+DOWN ARROW

To increase the height of the selected control

SHIFT+RIGHT ARROW

To increase the width of the selected control

SHIFT+UP ARROW

To reduce the height of the selected control

SHIFT+LEFT ARROW

To reduce the width of the selected control

[Window operations](#)

F11

To bring the Database window to the front

CTRL+F6

To cycle between open windows

ENTER

To restore the selected minimized window when all windows are minimized

CTRL+F8

To turn on Resize mode for the active window when it's not maximized;
press the arrow keys to resize the window

ALT+SPACEBAR

To display the Control menu

SHIFT+F10

To display the shortcut menu

CTRL+W or CTRL+F4

To close the active window

ALT+F11

To switch between the Visual Basic Editor and the previous active window

ALT+SHIFT+F11

To switch to the Microsoft Script Editor from the previous active window

[Working with Wizards](#)

TAB

Move to the Help  button in the wizard

ALT+N

To move to the next window in the wizard

ALT+B

To move to the previous window in the wizard

ALT+F

To close the wizard window

Note You can not use **CTRL+TAB** to navigate from one wizard window to another. **CTRL+TAB** will move the cursor between dialog boxes and controls on the wizard window.

[Miscellaneous](#)

F2

To display the complete hyperlink address for a selected hyperlink

F7

To check spelling

SHIFT+F2

To open the Zoom box to conveniently enter expressions and other text in small input areas

ALT+ENTER

To display a property sheet in Design view

ALT+F4

To quit Microsoft Access, close a dialog box, or close a property sheet

CTRL+F2

To invoke a Builder

CTRL+F11

To toggle between a custom menu bar and a built-in menu bar

CTRL+RIGHT ARROW or CTRL+COMMA (,)

To toggle forward between views when in a table, query, form, report, page, view, PivotTable list, PivotChart report, stored procedure, or Access project (.adp) function. If there are additional views available, successive keystrokes will move to the next available view.

CTRL+LEFT ARROW or CTRL+PERIOD (.)

To toggle back between views when in a table, query, form, report, page, view, PivotTable list, PivotChart report, stored procedure, or .adp function. If there are additional views available, successive keystrokes will move to the previous view

[Database window shortcut keys](#)

[Editing and navigating the Object list](#)

F2

To rename a selected object

DOWN ARROW

To move down one line

PAGE DOWN

To move down one window

END

To move to the last object

UP ARROW

To move up one line

PAGE UP

To move up one window

HOME

To move to the first object



[Navigating and opening objects](#)

CTRL+TAB

To cycle through the **Objects** bar top to bottom

SHIFT+CTRL+TAB

To cycle through the **Objects** bar bottom to top

ENTER or ALT+O

To open the selected table or query in Datasheet view, or form in Form view

ENTER

To open the selected report in Print Preview

ENTER

To open the selected data access page in Page view

ENTER

To run the selected macro

CTRL+ENTER or ALT+D

To open the selected table, query, form, report, data access page, macro, or module in Design view

ALT+N

To create a new table, query, form, report, data access page, macro, or module

F5

To refresh the Database window

CTRL+G

To display the Immediate window in the Visual Basic Editor



[Work with menus](#)

SHIFT+F10

To show the shortcut menu

F10

To make the menu bar active

ALT+SPACEBAR

To show the program icon menu (on the program title bar)

DOWN ARROW or UP ARROW

With the menu or submenu is visible, to select the next or previous command

LEFT ARROW or RIGHT ARROW

To select the menu to the left or right; or, when a submenu is visible, to switch between the main menu and the submenu

HOME or END

To select the first or last command on the menu or submenu

ALT

To close the visible menu and submenu at the same time

ESC

To close the visible menu; or, with a submenu visible, to close the submenu only



[Work with toolbars](#)

CTRL+TAB or CTRL+SHIFT+TAB

To select the next or previous toolbar

TAB or SHIFT+TAB

When a toolbar is active, to select the next or previous button or menu on the toolbar

ENTER

When a menu on a toolbar is selected, to open the menu

ENTER

When a button is selected, to perform the action assigned to a button



[Work in windows and dialog boxes](#)



[Using a program window](#)

ALT+TAB

To switch to the next program

ALT+SHIFT+TAB

To switch to the previous program

CTRL+ESC

To show the Windows **Start** menu

CTRL+W

To close the active database window

CTRL+F6

To switch to the next database window

CTRL+SHIFT+F6

To switch to the previous database window

ENTER

To restore the selected minimized window when all windows are minimized



[Using a dialog box](#)

CTRL+TAB or CTRL+PAGE DOWN

To switch to the next tab in a dialog box

CTRL+SHIFT+TAB or CTRL+PAGE UP

To switch to the previous tab in a dialog box

TAB

To move to the next option or option group

SHIFT+TAB

To move to the previous option or option group

Arrow keys

To move between options in the selected drop-down list box, or to move between some options in a group of options

SPACEBAR

To perform the action assigned to the selected button; select or clear the check box

Letter key for the first letter in the option name you want (when a drop-down list box is selected)

To move to the option by the first letter in the option name in a drop-down list box

ALT+letter key

To select the option, or to select or clear the check box by the letter underlined in the option name

ALT+DOWN ARROW

To open the selected drop-down list box

ESC

To close the selected drop-down list box

ENTER

To perform the action assigned to the default button in the dialog box
ESC

To cancel the command and close the dialog box

ALT+F4

To close a dialog box



[Editing in a text box](#)

HOME

To move to the beginning of the entry

END

To move to the end of the entry

LEFT ARROW or RIGHT ARROW

To move one character to the left or right

CTRL+LEFT ARROW or CTRL+RIGHT ARROW

To move one word to the left or right

SHIFT+HOME

To select from the insertion point to the beginning of the entry

SHIFT+END

To select from the insertion point to the end of the entry

SHIFT+LEFT ARROW

To select one character to the left

SHIFT+RIGHT ARROW

To select one character to the right

CTRL+SHIFT+LEFT ARROW

To select one word to the left

CTRL+SHIFT+RIGHT ARROW

To select one word to the right



[Work with the Open, File New Database, and Save dialog boxes](#)

ALT+1

Go to the previous folder ()

ALT+2

Open the folder up one level from the open folder (**Up One Level** button )

ALT+3

Close the dialog box, and open your World Wide WebWorld Wide Web

search page (**Search the Web** button )

ALT+4

Delete the selected folder or file (**Delete** button )

ALT+5

Create a new subfolder in the open folder (**Create New Folder** button )

ALT+6

Switch between List, Details, Properties, and Preview views

ALT+7

Show the **Tools** menu (**Tools** button)



[Work with task panes](#)

F6

Move to a [task pane](#) from another pane in the program window. (You may need to press F6 more than once.)

Note If pressing F6 doesn't display the task pane you want, try pressing ALT to place focus on the menu bar, and then pressing CTRL+TAB to move to the task pane.

CTRL+TAB

When a menu or [toolbar](#) is active, move to a task pane. (You may need to press CTRL+TAB more than once.)

TAB or SHIFT+TAB

When a task pane is active, select the next or previous option in the task pane

CTRL+DOWN ARROW

Display the full set of commands on the task pane menu

DOWN ARROW or UP ARROW

Move among choices in a selected submenu; move among certain options in a group of options

SPACEBAR or ENTER

Open the selected menu, or perform the action assigned to the selected button

SHIFT+F10

Open a [shortcut menu](#); open a drop-down menu for the selected gallery item

HOME or END

When a menu or submenu is visible, select the first or last command on the menu or submenu

PAGE UP or PAGE DOWN

Scroll up or down in the selected gallery list

CTRL+HOME or CTRL+END

Move to the top or bottom of the selected gallery list



[Keyboard shortcuts for using the **Help** task pane and Help window](#)

The Help Pane is a task pane that provides access to all Office Help content. As a task pane, the Help Pane appears as part of the active application. The Help window displays topics and other Help content and appears as a window next to, but separate from, the active application.

In the **Help** task pane

F1

Display the **Help** task pane.

F6

Switch between the **Help** task pane and the active application.

TAB

Select the next item in the **Help** task pane.

SHIFT+TAB

Select the previous item in the **Help** task pane.

ENTER

Perform the action for the selected item.

DOWN ARROW and UP ARROW

In a Table of Contents, select the next and previous item, respectively.

RIGHT ARROW and LEFT ARROW

In a Table of Contents, expand and collapse the selected item, respectively.

ALT+LEFT ARROW

Move back to the previous task Pane.

ALT+RIGHT ARROW

Move forward to the next task Pane.

CTRL+SPACEBAR

Open the menu of Pane options.

CTRL+F1

Close and reopen the current task pane.

RIGHT ARROW

Expand a +/- list.

LEFT ARROW

Collapse a +/- list.

In the Help window

TAB

Select the next hidden text or hyperlink, or **Show All** or **Hide All** at the top of a topic

SHIFT+TAB

Select the previous hidden text or hyperlink, or the **Browser View** button at the top of a Microsoft Office Web site article

ENTER

Perform the action for the selected **Show All**, **Hide All**, hidden text, or hyperlink

ALT+LEFT ARROW

Move back to the previous Help topic.

ALT+RIGHT ARROW

Move forward to the next Help topic.

CTRL+P

Print the current Help topic.

UP ARROW AND DOWN ARROW

Scroll small amounts up and down, respectively, within the currently-displayed Help topic.

PAGE UP AND PAGE DOWN

Scroll larger amounts up and down, respectively, within the currently-displayed Help topic.

ALT+U

Change whether the Help window appears connected to (tiled) or separate from (untiled) the active application.

SHIFT+F10

Display a menu of commands for the Help window; requires that the Help window have active focus (click an item in the Help window).



[Send e-mail messages](#)

ALT+F+D+A

Send the active database object as an e-mail message

CTRL+SHIFT+B

Open the Address Book

ALT+K

Check the names in the **To**, **Cc**, and **Bcc** boxes against the Address Book
TAB

Select the next box in the e-mail header or the body of the message when
the last box in the e-mail header is active

SHIFT+TAB

Select the previous field or button in the e-mail header

Keys for working with text and data



[Select text and data](#)



[Selecting text in a field](#)

SHIFT+RIGHT ARROW

To extend one character to the right

CTRL+SHIFT+RIGHT ARROW

To extend one word to the right

SHIFT+LEFT ARROW

To extend one character to the left

CTRL+SHIFT+LEFT ARROW

To extend one word to the left



[Selecting a field or record](#)

Note To cancel a selection, use the opposite arrow key.

TAB

To select the next field

F2

To switch between Edit mode (with insertion point displayed) and

[Navigation mode](#)

SHIFT+SPACEBAR

To switch between selecting the current record and the first field of the
current record, in Navigation mode

SHIFT+UP ARROW

To extend selection to the previous record, if the current record is selected

SHIFT+DOWN ARROW

To extend selection to the next record, if the current record is selected

CTRL+A or CTRL+SHIFT+SPACEBAR

To select all records

[Extending a selection](#)

F8

To turn on Extend mode (EXT appears in the lower-right corner of the window); pressing F8 repeatedly extends the selection to the word, the field, the record (in Datasheet view only), and all records

LEFT ARROW or RIGHT ARROW

To extend a selection to adjacent fields in the same row in Datasheet view

UP ARROW or DOWN ARROW

To extend a selection to adjacent rows in Datasheet view

SHIFT+F8

To undo the previous extension

ESC

To cancel Extend mode

[Selecting and moving a column in Datasheet view](#)

CTRL+SPACEBAR

To select the current column or cancel the column selection, in [Navigation mode](#) only

SHIFT+RIGHT ARROW

To select the column to the right, if the current column is selected

SHIFT+LEFT ARROW

To select the column to the left, if the current column is selected

CTRL+SHIFT+F8

To turn on [Move mode](#); then press the RIGHT ARROW or LEFT ARROW key to move selected column(s) to the right or left

[Edit text and data](#)

Note If the insertion point isn't visible, press F2 to display it.

[Moving the insertion point in a field](#)

RIGHT ARROW

To move it one character to the right

CTRL+RIGHT ARROW

To move it one word to the right

LEFT ARROW

To move it one character to the left

CTRL+LEFT ARROW

To move it one word to the left

END

To move it to the end of the field, in single-line fields

CTRL+END

To move it to the end of the field, in multiple-line fields

HOME

To move it to the beginning of the field, in single-line fields

CTRL+HOME

To move it to the beginning of the field, in multiple-line fields



[Copying, moving, or deleting text](#)

CTRL+C

To copy the selection to the Clipboard

CTRL+X

To cut the selection and copy it to the Clipboard

CTRL+V

To paste the contents of the Clipboard at the insertion point

BACKSPACE

To delete the selection or the character to the left of the insertion point

DELETE

To delete the selection or the character to the right of the insertion point

CTRL+DELETE

To delete all characters to the right of the insertion point



[Undoing changes](#)

CTRL+Z or ALT+BACKSPACE

To undo typing

ESC

To undo changes in the current field or current record; if both have been changed, press ESC twice to undo changes, first in the current field and then in the current record

[Entering data in Datasheet or Form view](#)

CTRL+SEMICOLON (;)

To insert the current date

CTRL+SHIFT+COLON (:)

To insert the current time

CTRL+ALT+SPACEBAR

To insert the default value for a field

CTRL+APOSTROPHE (')

To insert the value from the same field in the previous record

CTRL+PLUS SIGN (+)

To add a new record

CTRL+MINUS SIGN (-)

To delete the current record

SHIFT+ENTER

To save changes to the current record

SPACEBAR

To switch between the values in a check box or option button

CTRL+ENTER

To insert a new line

[Refreshing fields with current data](#)

F9

To recalculate the fields in the window

SHIFT+F9

To [query](#) the underlying tables; in a subform, this queries the underlying table for the subform only

F9

Refresh the contents of a [Lookup field](#) list box or combo box

Keys for navigating records

[Navigate in Design view](#)

F2

To switch between Edit mode (with insertion point displayed) and Navigation mode

- F4
To switch to the property sheet (Design view in forms and reports in databases and Access projects)
- F5
To switch to Form view from form Design view
- F6
To switch between the upper and lower portions of a window (Design view of the Advanced Filter/Sort window only)
- F7
To switch to the Code Builder from form or report Design view (Design view window or property sheet)
- F8
Invokes the field list in a form, report , or data access page. If field list is already open, moves focus to field list
- SHIFT+F7
To switch from the Visual Basic Editor to form or report Design view
- SHIFT+F7
To switch from a control's property sheet in form or report Design view to the design surface without changing the control focus
- ALT+ENTER
To display a property sheet in Design view
- ENTER or ALT+O
To open the selected form in Form view
- CTRL+ENTER or ALT+D
To open the selected table, query, form, report, data access page, macro, or module in Design view
- CTRL+TAB
With focus set to a section, will move the focus to a subsection
- SHIFT+ENTER
To add a control to a section
- CTRL+C
To copy the selected control to the Clipboard
- CTRL+X
To cut the selected control and copy it to the Clipboard
- CTRL+V
To paste the contents of the Clipboard in the upper-left corner of the selected section
- RIGHT ARROW
To move the selected control to the right by a pixel along the page's grid

LEFT ARROW

To move the selected control to the left by a pixel along the page's grid

UP ARROW

To move the selected control up by a pixel along the page's grid

DOWN ARROW

To move the selected control down by a pixel along the page's grid

CTRL+RIGHT ARROW

To move the selected control to the right by a pixel (irrespective of the page's grid)

CTRL+LEFT ARROW

To move the selected control to the left by a pixel (irrespective of the page's grid)

CTRL+UP ARROW

To move the selected control up by a pixel (irrespective of the page's grid)

CTRL+DOWN ARROW

To move the selected control down by a pixel (irrespective of the page's grid)

SHIFT+RIGHT ARROW

To increase the width of the selected control (to the right) by a pixel

SHIFT+LEFT ARROW

To decrease the width of the selected control (to the left) by a pixel

SHIFT+UP ARROW

To decrease the height of the selected control (from the bottom) by a pixel

SHIFT+DOWN ARROW

To increase the height of the selected control (from the bottom) by a pixel

SHIFT+TAB

To move the focus from the page design surface to the address/subject dialog box when sending a page as an e-mail

ESC

To move the focus from the Field List or Data Outline back to the data access page design surface



[Navigate in Datasheet view](#)



[Going to a specific record](#)

F5

To move to the [record number box](#); then type the record number and press

ENTER

[Navigating between fields and records](#)

TAB or RIGHT ARROW

To move to the next field

END

To move to the last field in the current record, in [Navigation mode](#)
SHIFT+TAB, or LEFT ARROW

To move to the previous field

HOME

To move to the first field in the current record, in Navigation mode
DOWN ARROW

To move to the current field in the next record

CTRL+DOWN ARROW

To move to the current field in the last record, in Navigation mode

CTRL+END

To move to the last field in the last record, in Navigation mode

UP ARROW

To move to the current field in the previous record

CTRL+UP ARROW

To move to the current field in the first record, in Navigation mode

CTRL+HOME

To move to the first field in the first record, in Navigation mode

[Navigating to another screen of data](#)

PAGE DOWN

To go down one screen

PAGE UP

To go up one screen

CTRL+PAGE DOWN

To go right one screen

CTRL+PAGE UP

To go left one screen

[Navigate in subdatasheets](#)

[Going to a specific record](#)

F5

From the subdatasheet to move to the [record number box](#); then type the record number and press ENTER

[Expanding and collapsing subdatasheet](#)

CTRL+SHIFT+DOWN ARROW

From the datasheet to expand the record's subdatasheet

CTRL+SHIFT+UP ARROW

To collapse the subdatasheet

[Navigating between the datasheet and subdatasheet](#)

TAB

To enter the subdatasheet from the last field of the previous record in the datasheet

SHIFT+TAB

To enter the subdatasheet from the first field of the following record in the datasheet

CTRL+TAB

To exit the subdatasheet and move to the first field of the next record in the datasheet

CTRL+SHIFT+TAB

To exit the subdatasheet and move to the last field of the previous record in the datasheet

TAB

From the last field in the subdatasheet to enter the next field in the datasheet

DOWN ARROW

From the datasheet to bypass the subdatasheet and move to the next record in the datasheet

UP ARROW

From the datasheet to bypass the subdatasheet and move to the previous record in the datasheet

Note You can navigate between fields and records in a subdatasheet with the same shortcut keys used in Datasheet view.

[Navigate in Form view](#)

[Going to a specific record](#)

F5

To move to the [record number box](#); then type the record number and press ENTER

[Navigating between fields and records](#)

TAB

To move to the next field

SHIFT+TAB

To move to the previous field

END

To move to the last field in the current record, in [Navigation mode](#)

CTRL+END

To move to the last field in the last record, in Navigation mode

HOME

To move to the first field in the current record, in Navigation mode

CTRL+HOME

To move to the first field in the first record, in Navigation mode

CTRL+PAGE DOWN

To move to the current field in the next record

CTRL+PAGE UP

To move to the current field in the previous record

[Navigating in forms with more than one page](#)

PAGE DOWN

To go down one page; at the end of the record, moves to the equivalent page on the next record

PAGE UP

To go up one page; at the end of the record, moves to the equivalent page on the previous record

[Navigating between the main form and subform](#)

TAB

To enter the subform from the preceding field in the main form

SHIFT+TAB

To enter the subform from the following field in the main form

CTRL+TAB

To exit the subform and move to the next field in the master form or next record

CTRL+SHIFT+TAB

To exit the subform and move to the previous field in the main form or previous record



[Navigate in Print Preview and Layout Preview](#)



[Dialog box and window operations](#)

P or CTRL+P

To open the **Print** dialog box

S

To open the **Page Setup** dialog box

Z

To zoom in or out on a part of the page

C or ESC

To cancel Print Preview or Layout Preview



[Viewing different pages](#)

F5

To move to the page number box; then type the page number and press ENTER

PAGE DOWN or DOWN ARROW

To view the next page (when **Fit To Window** is selected)

PAGE UP or UP ARROW

To view the previous page (when **Fit To Window** is selected)



[Navigating in Print and Layout Preview](#)

DOWN ARROW

To scroll down in small increments

PAGE DOWN

To scroll down one full screen

CTRL+DOWN ARROW

To move to the bottom of the page

UP ARROW

To scroll up in small increments

PAGE UP

To scroll up one full screen

CTRL+UP ARROW

To move to the top of the page

RIGHT ARROW

To scroll to the right in small increments

END or CTRL+RIGHT ARROW

To move to the right edge of the page

CTRL+END

To move to the lower-right corner of the page

LEFT ARROW

To scroll to the left in small increments

HOME or CTRL+LEFT ARROW

To move to the left edge of the page

CTRL+HOME

To move to the upper-left corner of the page



[Navigate in the Database Diagram window in a Microsoft Access project](#)

ESC

Move from a table cell to the table's title bar

ENTER

Move from a table's title bar to the last cell you edited

TAB

Move from table title bar to table title bar, or
from cell to cell inside a table

ALT + DOWN ARROW

Expand a list inside a table

DOWN ARROW

Scroll through the items in a drop-down list from top to bottom

UP ARROW

Move to the previous item in a list

ENTER

Select an item in a list and move to the next cell
SPACEBAR

Change the setting in a check box
HOME

Go to the first cell in the row, or
to the beginning of the current cell
END

Go to the last cell in the row, or
to the end of the current cell

PAGE DOWN
Scroll to the next "page" inside a table, or
to the next "page" of the diagram

PAGE UP
Scroll to the previous "page" inside a table, or
to the previous "page" of the diagram

[Navigate in the Query Designer in an Access project](#)

[Any Pane](#)

F6, SHIFT+F6
Move among the Query Designer panes

[Diagram Pane](#)

TAB, or SHIFT+TAB
Move among tables, views, and functions, (and to join lines, if available)

Arrow keys
Move between columns in a table, view, or function

SPACEBAR or PLUS key
Choose the selected data column for output

SPACEBAR or MINUS key
Remove the selected data column from the query output

DELETE
Remove the selected table, view, or function, or join line from the query

Note If multiple items are selected, pressing SPACEBAR affects all selected items. Select multiple items by holding down the SHIFT key while clicking

them. Toggle the selected state of a single item by holding down CTRL while clicking it.

Grid Pane

Arrow keys or TAB or SHIFT+TAB

Move among cells

CTRL+DOWN ARROW

Move to the last row in the current column

CTRL+UP ARROW

Move to the first row in the current column

CTRL+HOME

Move to the top left cell in the visible portion of grid

CTRL+END

Move to the bottom right cell

UP ARROW or DOWN ARROW

Move in a drop-down list

CTRL+SPACEBAR

Select an entire grid column

F2

Toggle between edit mode and cell selection mode

CTRL+C

Copy selected text in cell to the Clipboard (in edit mode)

CTRL+X

Cut selected text in cell and place it on the Clipboard (in edit mode)

CTRL+V

Paste text from the Clipboard (in edit mode)

INS

Toggle between insert and overstrike mode while editing in a cell

SPACEBAR

Toggle the check box in the Output column

Note If multiple items are selected, pressing this key affects all selected items.

DELETE

Clear the selected contents of a cell

DELETE

Remove row containing selected data column from the query

Note If multiple items are selected, pressing this key affects all selected items.

DELETE

Clear all values for a selected grid column

INS (after you select grid row)

Insert row between existing rows

INS (after you select any Or ... column)

Add an Or ... column

[SQL Pane](#)

You can use the standard Windows editing keys when working in the SQL pane, such as CTRL+Arrow keys to move between words, and the Cut, Copy, and Paste commands on the Edit menu.

Note You can only insert text; there is no overstrike mode.

[Navigate in Page view](#)

Note To use the following keys, press the TAB key repeatedly until the [expand indicator](#) or record navigation toolbar has the [focus](#). If a [PivotTable list](#) or [spreadsheet](#) control currently has the focus, press ESC.

SPACE or ENTER

Expand or collapse a group of records corresponding to the expand indicator in a [grouped data access page](#)

TAB or SHIFT+TAB

Select a button on the record navigation toolbar

SPACE or ENTER

Activate the selected button on the record navigation toolbar

[Work with PivotTable views](#)

[PivotTable view](#)

[Keys for selecting elements in PivotTable view](#)

The TAB key

Move the selection from left to right, and then down

ENTER

Move the selection from top to bottom, and then to the right

SHIFT+TAB

Select the cell to the left. If the current cell is the leftmost cell, SHIFT+TAB selects the last cell in the previous row.

SHIFT+ENTER

Select the cell above the current cell. If the current cell is the topmost cell, SHIFT+ENTER selects the last cell in the previous column.

CTRL+ENTER

Select the detail cells for the next item in the row area

SHIFT+CTRL+ENTER

Select the detail cells for the previous item in the row area

Arrow keys

Move the selection in the direction of the arrow key. If a row or column field is selected, press DOWN ARROW to move to the first item of data in the field, and then press an arrow key to move to the next or previous item or back to the field. If a detail field is selected, press DOWN ARROW or RIGHT ARROW to move to the first cell in the detail area.

SHIFT+arrow key

Extend or reduce the selection in the direction of the arrow key

CTRL+arrow key

Move the selection to the last cell in the direction of the arrow key

SHIFT+ALT+arrow key

Move the selected item in the direction of the arrow key

HOME

Select the leftmost cell of the current row

END

Select the rightmost cell of the current row

CTRL+HOME

Select the leftmost cell of the first row

CTRL+END

Select the last cell of the last row

SHIFT+CTRL+HOME

Extend selection to the leftmost cell of the first row

SHIFT+CTRL+END

Extend selection to the last cell of the last row

CTRL+SPACEBAR

Select the field for the currently selected item of data, total, or detail
SHIFT+SPACEBAR

Select the entire row containing the currently selected cell
CTRL+A

Select the entire [PivotTable view](#)
PAGE DOWN

Display the next screen
PAGE UP

Display the previous screen
SHIFT+PAGE DOWN

Extend a selection down one screen
SHIFT+PAGE UP

Reduce a selection by one screen
ALT+PAGE DOWN

Display the next screen to the right
ALT+PAGE UP

Display the previous screen to the left
SHIFT+ALT+PAGE DOWN

Extend selection to the page on the right
SHIFT+ALT+PAGE UP

Extend selection to the page on the left



[Keys for carrying out commands](#)

F1
Display Help topics.

SHIFT+F10
Display the shortcut menu for the selected element of the [PivotTable view](#).
Use the shortcut menus to carry out commands in the PivotTable view.

Underlined letter
Carry out a command on the shortcut menu

ESC
Close the shortcut menu without carrying out a command

ALT+ENTER
Display the **Properties** dialog box

ALT+F4
Close the **Properties** dialog box

ESC
Cancel a refresh operation in progress

CTRL+C

Copy the selected data from the PivotTable view to the Clipboard

CTRL+E

Export the contents of the PivotTable view to Microsoft Excel



[Keys for displaying, hiding, filtering, or sorting data](#)

CTRL+8

Show or hide the expand indicators (⊕ and ⊖ boxes) beside items

CTRL+PLUS SIGN

Expand the currently selected item

CTRL+MINUS SIGN

Hide the currently selected item

ALT+DOWN ARROW

Open the list for the currently selected field

The TAB key

Alternately move to the most recently selected item, the **OK** button, and the **Cancel** button in the drop-down list for a field

Arrow keys

Move to the next item in the drop-down list for a field

SPACEBAR

Select or clear the check box for the current item in the drop-down list for a field

ENTER

Close the drop-down list for a field and apply any changes you made

ESC

Close the drop-down list for a field without applying your changes

CTRL+T

Turn AutoFilter on or off

CTRL+SHIFT+A

Sort data in the selected field or total in ascending order (A – Z 0 – 9)

CTRL+SHIFT+Z

Sort data in the selected field or total in descending order (Z – A 9 – 0)

ALT+SHIFT+UP ARROW or ALT+SHIFT+LEFT ARROW

Move the selected member up or left

ALT+SHIFT+DOWN ARROW or ALT+SHIFT+RIGHT ARROW

Move the selected member down or right

 [Keys for adding fields and totals and changing the layout of a PivotTable view](#)

 [Keys for working with the field list](#)

CTRL+L

Display the field list, or activate it if it's already displayed

Arrow keys

Move to the next item in the field list

SHIFT+UP ARROW

Move to the previous item and include it in the selection

SHIFT+DOWN ARROW

Move to the next item and include it in the selection

CTRL+UP ARROW

Move to the previous item, but don't include the item in the selection

CTRL+DOWN ARROW

Move to the next item, but don't include the item in the selection

CTRL+SPACEBAR

Remove the item from the selection, if the item that has focus is included in the selection, and vice versa

PLUS SIGN (numeric keypad)

Expand the current item in the field list to display its contents. Or expand Totals to display the available total fields.

MINUS SIGN (numeric keypad)

Collapse the current item in the field list to hide its contents. Or collapse Totals to hide the available total fields.

The TAB key

Alternately move to the most recently selected item, the **Add to** button, and the list next to the **Add to** button in the field list

ALT+DOWN ARROW

Open the drop-down list next to the **Add to** button in the field list. Use the arrow keys to move to the next item in the list, and then press ENTER to select an item.

ENTER

Add the highlighted field in the field list to the area in the [PivotTable view](#) that's displayed in the **Add to** list

ALT+F4

Close the field list

[Keys for adding fields and totals](#)

CTRL+SHIFT+S

Add a new total field for the selected field in the [PivotTable view](#) by using the **Sum** summary function

CTRL+SHIFT+C

Add a new total field for the selected field in the PivotTable view by using the **Count** summary function

CTRL+SHIFT+M

Add a new total field for the selected field in the PivotTable view by using the **Min** summary function

CTRL+SHIFT+X

Add a new total field for the selected field in the PivotTable view by using the **Max** summary function

CTRL+SHIFT+E

Add a new total field for the selected field in the PivotTable view by using the **Average** summary function

CTRL+SHIFT+D

Add a new total field for the selected field in the PivotTable view by using the **Standard Deviation** summary function

CTRL+SHIFT+T

Add a new total field for the selected field in the PivotTable view by using the **Standard Deviation Population** summary function

CTRL+SHIFT+V

Add a new total field for the selected field in the PivotTable view by using the **Variance** summary function

CTRL+SHIFT+R

Add a new total field for the selected field in the PivotTable view by using the **Variance Population** summary function

CTRL+SHIFT+B

Turn subtotals and grand totals on or off for the selected field in the PivotTable view

CTRL+F

Add a calculated detail field

[Keys for changing the layout](#)

Note The following four shortcuts do not work if you press the keys 1, 2, 3, or 4

from the numeric pad of your keyboard.

CTRL+1

Move the selected field in the [PivotTable view](#) to the row area

CTRL+2

Move the selected field in the PivotTable view to the column area

CTRL+3

Move the selected field in the PivotTable view to the filter area

CTRL+4

Move the selected field in the PivotTable view to the detail area

CTRL+LEFT ARROW

Move the selected row or column field in the PivotTable view to a higher level

CTRL+RIGHT ARROW

Move the selected row or column field in the PivotTable view to a lower level



[Keys for formatting elements in PivotTable view](#)

To use the following shortcuts, first select a detail field or a data cell for a total field.

The first seven keyboard shortcuts change the number format of the selected field.

CTRL+SHIFT+~ (tilde)

Apply the general number format to values in the selected total or detail field

CTRL+SHIFT+\$

Apply the currency format, with two decimal places and negative numbers in parentheses, to values in the selected total or detail field

CTRL+SHIFT+%

Apply the percentage format, with no decimal places, to values in the selected total or detail field

CTRL+SHIFT+^

Apply the exponential number format, with two decimal places, to values in the selected total or detail field

CTRL+SHIFT+#

Apply the date format, with the day, month, and year, to values in the

selected total or detail field

CTRL+SHIFT+@

Apply the time format, with the hour, minute, and AM or PM, to values in the selected total or detail field

CTRL+SHIFT+!

Apply the numeric format, with two decimal places, thousands separator, and a minus sign for negative values, to values in the selected total or detail field

CTRL+B

Make text bold in the selected field of the [PivotTable view](#)

CTRL+U

Make text underlined in the selected field of the PivotTable view

CTRL+I

Make text italic in the selected field of the PivotTable view



[PivotChart view](#)



[Keys for selecting items in a chart](#)

RIGHT ARROW

Select the next item in the chart

LEFT ARROW

Select the previous item in the chart

DOWN ARROW

Select the next group of items

UP ARROW

Select the previous group of items



[Keys for working with properties and options](#)

ALT+ENTER

Display the **Properties** dialog box

ALT+F4

Close the **Properties** dialog box

The TAB key

When the **Properties** dialog box is active, select the next item on the active tab

RIGHT ARROW

When a tab in the **Properties** dialog box is active, select the next tab
LEFT ARROW

When a tab in the **Properties** dialog box is active, select the previous tab
DOWN ARROW

Display a list or palette when a button that contains a list or palette is selected

SHIFT+F10

Display the shortcut menu

Underlined letter

Carry out a command on the shortcut menu

ESC

Close the shortcut menu without carrying out a command



[Keys for working with fields](#)

ALT+DOWN ARROW

Open the list for the currently selected field

The TAB key

In the drop-down list for a field, alternately move to the most recently selected item, the **OK** button, and the **Cancel** button

Arrow keys

In the drop-down list for a field, move to the next item

SPACEBAR

In the drop-down list for a field, select or clear the check box for the current item

ENTER

Close the drop-down list for a field and apply any changes you made

ESC

Close the drop-down list for a field without applying your changes



[Keys for working with the field list](#)

CTRL+L

Display the field list, or activate it if it's already displayed

Arrow keys

Move to the next item in the field list

SHIFT+UP ARROW

Move to the previous item and include it in the selection

SHIFT+DOWN ARROW

Move to the next item and include it in the selection

CTRL+UP ARROW

Move to the previous item, but don't include the item in the selection

CTRL+DOWN ARROW

Move to the next item, but don't include the item in the selection

CTRL+SPACEBAR

Remove the item from the selection if the item that has focus is included in the selection, and vice versa

PLUS SIGN (numeric keypad)

Expand the current item in the field list to display its contents, or expand Totals to display the available total fields

MINUS SIGN (numeric keypad)

Collapse the current item in the field list to hide its contents, or collapse Totals to hide the available total fields.

The TAB key

In the field list, alternately move to the most recently selected item, the **Add to** button, and the list next to the **Add to** button

ALT+DOWN ARROW

Open the drop-down list next to the **Add to** button in the field list. Use the arrow keys to move to the next item in the list, and then press ENTER to select an item.

ENTER

Add the highlighted field in the field list to the drop area that's displayed in the **Add to** list

ALT+F4

Close the field list

[Show All](#)

Keyboard Shortcuts

You can use shortcut keys for quick access to frequently used commands or operations. The topics below list the shortcut keys available in Microsoft Access. You can also use access keys to move the focus to a menu, command, or control without using the mouse.

General shortcut keys

[Global Access shortcut keys](#)

[Opening databases](#)

CTRL+N

To open a new database

CTRL+O

To open an existing database

ALT+F4

To quit Microsoft Access

[Printing and saving](#)

CTRL+P

To print the current or selected object

P or CTRL+P

To open the **Print** dialog box

S

To open the **Page Setup** dialog box

C or ESC

To cancel Print Preview or Layout Preview

CTRL+S or SHIFT+F12 or ALT+SHIFT+F2

To save a database object

F12 or ALT+F2

To open the **Save As** dialog box

[Using a combo box or list box](#)

F4 or ALT+DOWN ARROW

To open a combo box

F9

To refresh the contents of a [Lookup field](#) list box or combo box

DOWN ARROW

To move down one line

PAGE DOWN

To move down one page

UP ARROW

To move up one line

PAGE UP

To move up one page

TAB

To exit the combo box or list box



[Finding and replacing text or data](#)

CTRL+F

To open the **Find** tab in the **Find and Replace** dialog box (Datasheet view and Form view only)

CTRL+H

To open the **Replace** tab in the **Find and Replace** dialog box (Datasheet view and Form view only)

SHIFT+F4

To find the next occurrence of the text specified in the **Find and Replace** dialog box when the dialog box is closed (Datasheet view and Form view only)



[Working in Design view](#)

F2

To switch between Edit mode (with insertion point displayed) and [Navigation mode](#)

F4

To switch to the property sheet (Design view in forms and reports in databases and Access projects)

F5

To switch to Form view from form Design view

F6

To switch between the upper and lower portions of a window (Design view of tables, macros, and queries and the Advanced Filter/Sort window only)

F7

To switch to the Code Builder from form or report Design view (Design view window or property sheet)

SHIFT+F7

To switch from the Visual Basic Editor to form or report Design view

ALT+V+P

To open property sheet for a selected object



[Editing controls in form and report Design view](#)

SHIFT+ENTER

To add a control to a section

CTRL+C

To copy the selected control to the Clipboard

CTRL+X

To cut the selected control and copy it to the Clipboard

CTRL+V

To paste the contents of the Clipboard in the upper-left corner of the selected section

CTRL+RIGHT ARROW

To move the selected control to the right

CTRL+LEFT ARROW

To move the selected control to the left

CTRL+UP ARROW

To move the selected control up

CTRL+DOWN ARROW

To move the selected control down

SHIFT+DOWN ARROW

To increase the height of the selected control

SHIFT+RIGHT ARROW

To increase the width of the selected control

SHIFT+UP ARROW

To reduce the height of the selected control

SHIFT+LEFT ARROW

To reduce the width of the selected control

[Window operations](#)

F11

To bring the Database window to the front

CTRL+F6

To cycle between open windows

ENTER

To restore the selected minimized window when all windows are minimized

CTRL+F8

To turn on Resize mode for the active window when it's not maximized;
press the arrow keys to resize the window

ALT+SPACEBAR

To display the Control menu

SHIFT+F10

To display the shortcut menu

CTRL+W or CTRL+F4

To close the active window

ALT+F11

To switch between the Visual Basic Editor and the previous active window

ALT+SHIFT+F11

To switch to the Microsoft Script Editor from the previous active window

[Working with Wizards](#)

TAB

Move to the Help  button in the wizard

ALT+N

To move to the next window in the wizard

ALT+B

To move to the previous window in the wizard

ALT+F

To close the wizard window

Note You can not use **CTRL+TAB** to navigate from one wizard window to another. **CTRL+TAB** will move the cursor between dialog boxes and controls on the wizard window.

[Miscellaneous](#)

F2

To display the complete hyperlink address for a selected hyperlink

F7

To check spelling

SHIFT+F2

To open the Zoom box to conveniently enter expressions and other text in small input areas

ALT+ENTER

To display a property sheet in Design view

ALT+F4

To quit Microsoft Access, close a dialog box, or close a property sheet

CTRL+F2

To invoke a Builder

CTRL+F11

To toggle between a custom menu bar and a built-in menu bar

CTRL+RIGHT ARROW or CTRL+COMMA (,)

To toggle forward between views when in a table, query, form, report, page, view, PivotTable list, PivotChart report, stored procedure, or Access project (.adp) function. If there are additional views available, successive keystrokes will move to the next available view.

CTRL+LEFT ARROW or CTRL+PERIOD (.)

To toggle back between views when in a table, query, form, report, page, view, PivotTable list, PivotChart report, stored procedure, or .adp function. If there are additional views available, successive keystrokes will move to the previous view



[Database window shortcut keys](#)



[Editing and navigating the Object list](#)

F2

To rename a selected object

DOWN ARROW

To move down one line

PAGE DOWN

To move down one window

END

To move to the last object

UP ARROW

To move up one line

PAGE UP

To move up one window

HOME

To move to the first object



[Navigating and opening objects](#)

CTRL+TAB

To cycle through the **Objects** bar top to bottom

SHIFT+CTRL+TAB

To cycle through the **Objects** bar bottom to top

ENTER or ALT+O

To open the selected table or query in Datasheet view, or form in Form view

ENTER

To open the selected report in Print Preview

ENTER

To open the selected data access page in Page view

ENTER

To run the selected macro

CTRL+ENTER or ALT+D

To open the selected table, query, form, report, data access page, macro, or module in Design view

ALT+N

To create a new table, query, form, report, data access page, macro, or module

F5

To refresh the Database window

CTRL+G

To display the Immediate window in the Visual Basic Editor



[Work with menus](#)

SHIFT+F10

To show the shortcut menu

F10

To make the menu bar active

ALT+SPACEBAR

To show the program icon menu (on the program title bar)

DOWN ARROW or UP ARROW

With the menu or submenu is visible, to select the next or previous command

LEFT ARROW or RIGHT ARROW

To select the menu to the left or right; or, when a submenu is visible, to switch between the main menu and the submenu

HOME or END

To select the first or last command on the menu or submenu

ALT

To close the visible menu and submenu at the same time

ESC

To close the visible menu; or, with a submenu visible, to close the submenu only



[Work with toolbars](#)

CTRL+TAB or CTRL+SHIFT+TAB

To select the next or previous toolbar

TAB or SHIFT+TAB

When a toolbar is active, to select the next or previous button or menu on the toolbar

ENTER

When a menu on a toolbar is selected, to open the menu

ENTER

When a button is selected, to perform the action assigned to a button



[Work in windows and dialog boxes](#)



[Using a program window](#)

ALT+TAB

To switch to the next program

ALT+SHIFT+TAB

To switch to the previous program

CTRL+ESC

To show the Windows **Start** menu

CTRL+W

To close the active database window

CTRL+F6

To switch to the next database window

CTRL+SHIFT+F6

To switch to the previous database window

ENTER

To restore the selected minimized window when all windows are minimized



[Using a dialog box](#)

CTRL+TAB or CTRL+PAGE DOWN

To switch to the next tab in a dialog box

CTRL+SHIFT+TAB or CTRL+PAGE UP

To switch to the previous tab in a dialog box

TAB

To move to the next option or option group

SHIFT+TAB

To move to the previous option or option group

Arrow keys

To move between options in the selected drop-down list box, or to move between some options in a group of options

SPACEBAR

To perform the action assigned to the selected button; select or clear the check box

Letter key for the first letter in the option name you want (when a drop-down list box is selected)

To move to the option by the first letter in the option name in a drop-down list box

ALT+letter key

To select the option, or to select or clear the check box by the letter underlined in the option name

ALT+DOWN ARROW

To open the selected drop-down list box

ESC

To close the selected drop-down list box

ENTER

To perform the action assigned to the default button in the dialog box

ESC

To cancel the command and close the dialog box
ALT+F4

To close a dialog box

[Editing in a text box](#)

HOME

To move to the beginning of the entry

END

To move to the end of the entry

LEFT ARROW or RIGHT ARROW

To move one character to the left or right

CTRL+LEFT ARROW or CTRL+RIGHT ARROW

To move one word to the left or right

SHIFT+HOME

To select from the insertion point to the beginning of the entry

SHIFT+END

To select from the insertion point to the end of the entry

SHIFT+LEFT ARROW

To select one character to the left

SHIFT+RIGHT ARROW

To select one character to the right

CTRL+SHIFT+LEFT ARROW

To select one word to the left

CTRL+SHIFT+RIGHT ARROW

To select one word to the right

[Work with the Open, File New Database, and Save dialog boxes](#)

ALT+1

Go to the previous folder ()

ALT+2

Open the folder up one level from the open folder (**Up One Level** button )

ALT+3

Close the dialog box, and open your World Wide WebWorld Wide Web search page (**Search the Web** button )

ALT+4

- Delete the selected folder or file (**Delete** button )
- ALT+5
 - Create a new subfolder in the open folder (**Create New Folder** button )
- ALT+6
 - Switch between List, Details, Properties, and Preview views
- ALT+7
 - Show the **Tools** menu (**Tools** button)

[Work with task panes](#)

- F6
 - Move to a [task pane](#) from another pane in the program window. (You may need to press F6 more than once.)

Note If pressing F6 doesn't display the task pane you want, try pressing ALT to place focus on the menu bar, and then pressing CTRL+TAB to move to the task pane.

- CTRL+TAB
 - When a menu or [toolbar](#) is active, move to a task pane. (You may need to press CTRL+TAB more than once.)
- TAB or SHIFT+TAB
 - When a task pane is active, select the next or previous option in the task pane
- CTRL+DOWN ARROW
 - Display the full set of commands on the task pane menu
- DOWN ARROW or UP ARROW
 - Move among choices in a selected submenu; move among certain options in a group of options
- SPACEBAR or ENTER
 - Open the selected menu, or perform the action assigned to the selected button
- SHIFT+F10
 - Open a [shortcut menu](#); open a drop-down menu for the selected gallery item
- HOME or END
 - When a menu or submenu is visible, select the first or last command on the menu or submenu
- PAGE UP or PAGE DOWN
 - Scroll up or down in the selected gallery list

CTRL+HOME or CTRL+END

Move to the top or bottom of the selected gallery list



[Keyboard shortcuts for using the **Help** task pane and Help window](#)

The Help Pane is a task pane that provides access to all Office Help content. As a task pane, the Help Pane appears as part of the active application. The Help window displays topics and other Help content and appears as a window next to, but separate from, the active application.

In the **Help** task pane

F1

Display the **Help** task pane.

F6

Switch between the **Help** task pane and the active application.

TAB

Select the next item in the **Help** task pane.

SHIFT+TAB

Select the previous item in the **Help** task pane.

ENTER

Perform the action for the selected item.

DOWN ARROW and UP ARROW

In a Table of Contents, select the next and previous item, respectively.

RIGHT ARROW and LEFT ARROW

In a Table of Contents, expand and collapse the selected item, respectively.

ALT+LEFT ARROW

Move back to the previous task Pane.

ALT+RIGHT ARROW

Move forward to the next task Pane.

CTRL+SPACEBAR

Open the menu of Pane options.

CTRL+F1

Close and reopen the current task pane.

RIGHT ARROW

Expand a +/- list.

LEFT ARROW

Collapse a +/- list.

In the Help window

TAB

Select the next hidden text or hyperlink, or **Show All** or **Hide All** at the top of a topic

SHIFT+TAB

Select the previous hidden text or hyperlink, or the **Browser View** button at the top of a Microsoft Office Web site article

ENTER

Perform the action for the selected **Show All**, **Hide All**, hidden text, or hyperlink

ALT+LEFT ARROW

Move back to the previous Help topic.

ALT+RIGHT ARROW

Move forward to the next Help topic.

CTRL+P

Print the current Help topic.

UP ARROW AND DOWN ARROW

Scroll small amounts up and down, respectively, within the currently-displayed Help topic.

PAGE UP AND PAGE DOWN

Scroll larger amounts up and down, respectively, within the currently-displayed Help topic.

ALT+U

Change whether the Help window appears connected to (tiled) or separate from (untiled) the active application.

SHIFT+F10

Display a menu of commands for the Help window; requires that the Help window have active focus (click an item in the Help window).



[Send e-mail messages](#)

ALT+F+D+A

Send the active database object as an e-mail message

CTRL+SHIFT+B

Open the Address Book

ALT+K

Check the names in the **To**, **Cc**, and **Bcc** boxes against the Address Book
TAB

Select the next box in the e-mail header or the body of the message when
the last box in the e-mail header is active

SHIFT+TAB

Select the previous field or button in the e-mail header

Keys for working with text and data



[Select text and data](#)



[Selecting text in a field](#)

SHIFT+RIGHT ARROW

To extend one character to the right

CTRL+SHIFT+RIGHT ARROW

To extend one word to the right

SHIFT+LEFT ARROW

To extend one character to the left

CTRL+SHIFT+LEFT ARROW

To extend one word to the left



[Selecting a field or record](#)

Note To cancel a selection, use the opposite arrow key.

TAB

To select the next field

F2

To switch between Edit mode (with insertion point displayed) and

[Navigation mode](#)

SHIFT+SPACEBAR

To switch between selecting the current record and the first field of the
current record, in Navigation mode

SHIFT+UP ARROW

To extend selection to the previous record, if the current record is selected

SHIFT+DOWN ARROW

To extend selection to the next record, if the current record is selected

CTRL+A or CTRL+SHIFT+SPACEBAR

To select all records

[Extending a selection](#)

F8

To turn on Extend mode (EXT appears in the lower-right corner of the window); pressing F8 repeatedly extends the selection to the word, the field, the record (in Datasheet view only), and all records

LEFT ARROW or RIGHT ARROW

To extend a selection to adjacent fields in the same row in Datasheet view

UP ARROW or DOWN ARROW

To extend a selection to adjacent rows in Datasheet view

SHIFT+F8

To undo the previous extension

ESC

To cancel Extend mode

[Selecting and moving a column in Datasheet view](#)

CTRL+SPACEBAR

To select the current column or cancel the column selection, in [Navigation mode](#) only

SHIFT+RIGHT ARROW

To select the column to the right, if the current column is selected

SHIFT+LEFT ARROW

To select the column to the left, if the current column is selected

CTRL+SHIFT+F8

To turn on [Move mode](#); then press the RIGHT ARROW or LEFT ARROW key to move selected column(s) to the right or left

[Edit text and data](#)

Note If the insertion point isn't visible, press F2 to display it.

[Moving the insertion point in a field](#)

RIGHT ARROW

To move it one character to the right

CTRL+RIGHT ARROW

To move it one word to the right

LEFT ARROW

To move it one character to the left

CTRL+LEFT ARROW

To move it one word to the left

END

To move it to the end of the field, in single-line fields

CTRL+END

To move it to the end of the field, in multiple-line fields

HOME

To move it to the beginning of the field, in single-line fields

CTRL+HOME

To move it to the beginning of the field, in multiple-line fields



[Copying, moving, or deleting text](#)

CTRL+C

To copy the selection to the Clipboard

CTRL+X

To cut the selection and copy it to the Clipboard

CTRL+V

To paste the contents of the Clipboard at the insertion point

BACKSPACE

To delete the selection or the character to the left of the insertion point

DELETE

To delete the selection or the character to the right of the insertion point

CTRL+DELETE

To delete all characters to the right of the insertion point



[Undoing changes](#)

CTRL+Z or ALT+BACKSPACE

To undo typing

ESC

To undo changes in the current field or current record; if both have been changed, press ESC twice to undo changes, first in the current field and then in the current record

[Entering data in Datasheet or Form view](#)

CTRL+SEMICOLON (;)

To insert the current date

CTRL+SHIFT+COLON (:)

To insert the current time

CTRL+ALT+SPACEBAR

To insert the default value for a field

CTRL+APOSTROPHE (')

To insert the value from the same field in the previous record

CTRL+PLUS SIGN (+)

To add a new record

CTRL+MINUS SIGN (-)

To delete the current record

SHIFT+ENTER

To save changes to the current record

SPACEBAR

To switch between the values in a check box or option button

CTRL+ENTER

To insert a new line

[Refreshing fields with current data](#)

F9

To recalculate the fields in the window

SHIFT+F9

To [query](#) the underlying tables; in a subform, this queries the underlying table for the subform only

F9

Refresh the contents of a [Lookup field](#) list box or combo box

Keys for navigating records

[Navigate in Design view](#)

F2

To switch between Edit mode (with insertion point displayed) and Navigation mode

- F4
To switch to the property sheet (Design view in forms and reports in databases and Access projects)
- F5
To switch to Form view from form Design view
- F6
To switch between the upper and lower portions of a window (Design view of the Advanced Filter/Sort window only)
- F7
To switch to the Code Builder from form or report Design view (Design view window or property sheet)
- F8
Invokes the field list in a form, report, or data access page. If field list is already open, moves focus to field list
- SHIFT+F7
To switch from the Visual Basic Editor to form or report Design view
- SHIFT+F7
To switch from a control's property sheet in form or report Design view to the design surface without changing the control focus
- ALT+ENTER
To display a property sheet in Design view
- ENTER or ALT+O
To open the selected form in Form view
- CTRL+ENTER or ALT+D
To open the selected table, query, form, report, data access page, macro, or module in Design view
- CTRL+TAB
With focus set to a section, will move the focus to a subsection
- SHIFT+ENTER
To add a control to a section
- CTRL+C
To copy the selected control to the Clipboard
- CTRL+X
To cut the selected control and copy it to the Clipboard
- CTRL+V
To paste the contents of the Clipboard in the upper-left corner of the selected section
- RIGHT ARROW
To move the selected control to the right by a pixel along the page's grid

LEFT ARROW

To move the selected control to the left by a pixel along the page's grid

UP ARROW

To move the selected control up by a pixel along the page's grid

DOWN ARROW

To move the selected control down by a pixel along the page's grid

CTRL+RIGHT ARROW

To move the selected control to the right by a pixel (irrespective of the page's grid)

CTRL+LEFT ARROW

To move the selected control to the left by a pixel (irrespective of the page's grid)

CTRL+UP ARROW

To move the selected control up by a pixel (irrespective of the page's grid)

CTRL+DOWN ARROW

To move the selected control down by a pixel (irrespective of the page's grid)

SHIFT+RIGHT ARROW

To increase the width of the selected control (to the right) by a pixel

SHIFT+LEFT ARROW

To decrease the width of the selected control (to the left) by a pixel

SHIFT+UP ARROW

To decrease the height of the selected control (from the bottom) by a pixel

SHIFT+DOWN ARROW

To increase the height of the selected control (from the bottom) by a pixel

SHIFT+TAB

To move the focus from the page design surface to the address/subject dialog box when sending a page as an e-mail

ESC

To move the focus from the Field List or Data Outline back to the data access page design surface



[Navigate in Datasheet view](#)



[Going to a specific record](#)

F5

To move to the [record number box](#); then type the record number and press

ENTER

[Navigating between fields and records](#)

TAB or RIGHT ARROW

To move to the next field

END

To move to the last field in the current record, in [Navigation mode](#)
SHIFT+TAB, or LEFT ARROW

To move to the previous field

HOME

To move to the first field in the current record, in Navigation mode
DOWN ARROW

To move to the current field in the next record

CTRL+DOWN ARROW

To move to the current field in the last record, in Navigation mode

CTRL+END

To move to the last field in the last record, in Navigation mode

UP ARROW

To move to the current field in the previous record

CTRL+UP ARROW

To move to the current field in the first record, in Navigation mode

CTRL+HOME

To move to the first field in the first record, in Navigation mode

[Navigating to another screen of data](#)

PAGE DOWN

To go down one screen

PAGE UP

To go up one screen

CTRL+PAGE DOWN

To go right one screen

CTRL+PAGE UP

To go left one screen

[Navigate in subdatasheets](#)

[Going to a specific record](#)

F5

From the subdatasheet to move to the [record number box](#); then type the record number and press ENTER

[Expanding and collapsing subdatasheet](#)

CTRL+SHIFT+DOWN ARROW

From the datasheet to expand the record's subdatasheet

CTRL+SHIFT+UP ARROW

To collapse the subdatasheet

[Navigating between the datasheet and subdatasheet](#)

TAB

To enter the subdatasheet from the last field of the previous record in the datasheet

SHIFT+TAB

To enter the subdatasheet from the first field of the following record in the datasheet

CTRL+TAB

To exit the subdatasheet and move to the first field of the next record in the datasheet

CTRL+SHIFT+TAB

To exit the subdatasheet and move to the last field of the previous record in the datasheet

TAB

From the last field in the subdatasheet to enter the next field in the datasheet

DOWN ARROW

From the datasheet to bypass the subdatasheet and move to the next record in the datasheet

UP ARROW

From the datasheet to bypass the subdatasheet and move to the previous record in the datasheet

Note You can navigate between fields and records in a subdatasheet with the same shortcut keys used in Datasheet view.

[Navigate in Form view](#)

[Going to a specific record](#)

F5

To move to the [record number box](#); then type the record number and press ENTER

[Navigating between fields and records](#)

TAB

To move to the next field

SHIFT+TAB

To move to the previous field

END

To move to the last field in the current record, in [Navigation mode](#)

CTRL+END

To move to the last field in the last record, in Navigation mode

HOME

To move to the first field in the current record, in Navigation mode

CTRL+HOME

To move to the first field in the first record, in Navigation mode

CTRL+PAGE DOWN

To move to the current field in the next record

CTRL+PAGE UP

To move to the current field in the previous record

[Navigating in forms with more than one page](#)

PAGE DOWN

To go down one page; at the end of the record, moves to the equivalent page on the next record

PAGE UP

To go up one page; at the end of the record, moves to the equivalent page on the previous record

[Navigating between the main form and subform](#)

TAB

To enter the subform from the preceding field in the main form

SHIFT+TAB

To enter the subform from the following field in the main form

CTRL+TAB

To exit the subform and move to the next field in the master form or next record

CTRL+SHIFT+TAB

To exit the subform and move to the previous field in the main form or previous record



[Navigate in Print Preview and Layout Preview](#)



[Dialog box and window operations](#)

P or CTRL+P

To open the **Print** dialog box

S

To open the **Page Setup** dialog box

Z

To zoom in or out on a part of the page

C or ESC

To cancel Print Preview or Layout Preview



[Viewing different pages](#)

F5

To move to the page number box; then type the page number and press ENTER

PAGE DOWN or DOWN ARROW

To view the next page (when **Fit To Window** is selected)

PAGE UP or UP ARROW

To view the previous page (when **Fit To Window** is selected)



[Navigating in Print and Layout Preview](#)

DOWN ARROW

To scroll down in small increments

PAGE DOWN

To scroll down one full screen

CTRL+DOWN ARROW

To move to the bottom of the page

UP ARROW

To scroll up in small increments

PAGE UP

To scroll up one full screen

CTRL+UP ARROW

To move to the top of the page

RIGHT ARROW

To scroll to the right in small increments

END or CTRL+RIGHT ARROW

To move to the right edge of the page

CTRL+END

To move to the lower-right corner of the page

LEFT ARROW

To scroll to the left in small increments

HOME or CTRL+LEFT ARROW

To move to the left edge of the page

CTRL+HOME

To move to the upper-left corner of the page



[Navigate in the Database Diagram window in a Microsoft Access project](#)

ESC

Move from a table cell to the table's title bar

ENTER

Move from a table's title bar to the last cell you edited

TAB

Move from table title bar to table title bar, or
from cell to cell inside a table

ALT + DOWN ARROW

Expand a list inside a table

DOWN ARROW

Scroll through the items in a drop-down list from top to bottom

UP ARROW

Move to the previous item in a list

ENTER

Select an item in a list and move to the next cell
SPACEBAR

Change the setting in a check box
HOME

Go to the first cell in the row, or
to the beginning of the current cell
END

Go to the last cell in the row, or
to the end of the current cell

PAGE DOWN
Scroll to the next "page" inside a table, or
to the next "page" of the diagram

PAGE UP
Scroll to the previous "page" inside a table, or
to the previous "page" of the diagram

[Navigate in the Query Designer in an Access project](#)

[Any Pane](#)

F6, SHIFT+F6
Move among the Query Designer panes

[Diagram Pane](#)

TAB, or SHIFT+TAB
Move among tables, views, and functions, (and to join lines, if available)

Arrow keys
Move between columns in a table, view, or function

SPACEBAR or PLUS key
Choose the selected data column for output

SPACEBAR or MINUS key
Remove the selected data column from the query output

DELETE
Remove the selected table, view, or function, or join line from the query

Note If multiple items are selected, pressing SPACEBAR affects all selected items. Select multiple items by holding down the SHIFT key while clicking

them. Toggle the selected state of a single item by holding down CTRL while clicking it.

Grid Pane

Arrow keys or TAB or SHIFT+TAB

Move among cells

CTRL+DOWN ARROW

Move to the last row in the current column

CTRL+UP ARROW

Move to the first row in the current column

CTRL+HOME

Move to the top left cell in the visible portion of grid

CTRL+END

Move to the bottom right cell

UP ARROW or DOWN ARROW

Move in a drop-down list

CTRL+SPACEBAR

Select an entire grid column

F2

Toggle between edit mode and cell selection mode

CTRL+C

Copy selected text in cell to the Clipboard (in edit mode)

CTRL+X

Cut selected text in cell and place it on the Clipboard (in edit mode)

CTRL+V

Paste text from the Clipboard (in edit mode)

INS

Toggle between insert and overstrike mode while editing in a cell

SPACEBAR

Toggle the check box in the Output column

Note If multiple items are selected, pressing this key affects all selected items.

DELETE

Clear the selected contents of a cell

DELETE

Remove row containing selected data column from the query

Note If multiple items are selected, pressing this key affects all selected items.

DELETE

Clear all values for a selected grid column

INS (after you select grid row)

Insert row between existing rows

INS (after you select any Or ... column)

Add an Or ... column

[SQL Pane](#)

You can use the standard Windows editing keys when working in the SQL pane, such as CTRL+Arrow keys to move between words, and the Cut, Copy, and Paste commands on the Edit menu.

Note You can only insert text; there is no overstrike mode.

[Navigate in Page view](#)

Note To use the following keys, press the TAB key repeatedly until the [expand indicator](#) or record navigation toolbar has the [focus](#). If a [PivotTable list](#) or [spreadsheet](#) control currently has the focus, press ESC.

SPACE or ENTER

Expand or collapse a group of records corresponding to the expand indicator in a [grouped data access page](#)

TAB or SHIFT+TAB

Select a button on the record navigation toolbar

SPACE or ENTER

Activate the selected button on the record navigation toolbar

[Work with PivotTable views](#)

[PivotTable view](#)

[Keys for selecting elements in PivotTable view](#)

The TAB key

Move the selection from left to right, and then down

ENTER

Move the selection from top to bottom, and then to the right

SHIFT+TAB

Select the cell to the left. If the current cell is the leftmost cell, SHIFT+TAB selects the last cell in the previous row.

SHIFT+ENTER

Select the cell above the current cell. If the current cell is the topmost cell, SHIFT+ENTER selects the last cell in the previous column.

CTRL+ENTER

Select the detail cells for the next item in the row area

SHIFT+CTRL+ENTER

Select the detail cells for the previous item in the row area

Arrow keys

Move the selection in the direction of the arrow key. If a row or column field is selected, press DOWN ARROW to move to the first item of data in the field, and then press an arrow key to move to the next or previous item or back to the field. If a detail field is selected, press DOWN ARROW or RIGHT ARROW to move to the first cell in the detail area.

SHIFT+arrow key

Extend or reduce the selection in the direction of the arrow key

CTRL+arrow key

Move the selection to the last cell in the direction of the arrow key

SHIFT+ALT+arrow key

Move the selected item in the direction of the arrow key

HOME

Select the leftmost cell of the current row

END

Select the rightmost cell of the current row

CTRL+HOME

Select the leftmost cell of the first row

CTRL+END

Select the last cell of the last row

SHIFT+CTRL+HOME

Extend selection to the leftmost cell of the first row

SHIFT+CTRL+END

Extend selection to the last cell of the last row

CTRL+SPACEBAR

Select the field for the currently selected item of data, total, or detail
SHIFT+SPACEBAR

Select the entire row containing the currently selected cell
CTRL+A

Select the entire [PivotTable view](#)
PAGE DOWN

Display the next screen
PAGE UP

Display the previous screen
SHIFT+PAGE DOWN

Extend a selection down one screen
SHIFT+PAGE UP

Reduce a selection by one screen
ALT+PAGE DOWN

Display the next screen to the right
ALT+PAGE UP

Display the previous screen to the left
SHIFT+ALT+PAGE DOWN

Extend selection to the page on the right
SHIFT+ALT+PAGE UP

Extend selection to the page on the left



[Keys for carrying out commands](#)

F1
Display Help topics.

SHIFT+F10
Display the shortcut menu for the selected element of the [PivotTable view](#).
Use the shortcut menus to carry out commands in the PivotTable view.

Underlined letter
Carry out a command on the shortcut menu

ESC
Close the shortcut menu without carrying out a command

ALT+ENTER
Display the **Properties** dialog box

ALT+F4
Close the **Properties** dialog box

ESC
Cancel a refresh operation in progress

CTRL+C

Copy the selected data from the PivotTable view to the Clipboard

CTRL+E

Export the contents of the PivotTable view to Microsoft Excel



[Keys for displaying, hiding, filtering, or sorting data](#)

CTRL+8

Show or hide the expand indicators (⊕ and ⊖ boxes) beside items

CTRL+PLUS SIGN

Expand the currently selected item

CTRL+MINUS SIGN

Hide the currently selected item

ALT+DOWN ARROW

Open the list for the currently selected field

The TAB key

Alternately move to the most recently selected item, the **OK** button, and the **Cancel** button in the drop-down list for a field

Arrow keys

Move to the next item in the drop-down list for a field

SPACEBAR

Select or clear the check box for the current item in the drop-down list for a field

ENTER

Close the drop-down list for a field and apply any changes you made

ESC

Close the drop-down list for a field without applying your changes

CTRL+T

Turn AutoFilter on or off

CTRL+SHIFT+A

Sort data in the selected field or total in ascending order (A – Z 0 – 9)

CTRL+SHIFT+Z

Sort data in the selected field or total in descending order (Z – A 9 – 0)

ALT+SHIFT+UP ARROW or ALT+SHIFT+LEFT ARROW

Move the selected member up or left

ALT+SHIFT+DOWN ARROW or ALT+SHIFT+RIGHT ARROW

Move the selected member down or right

 [Keys for adding fields and totals and changing the layout of a PivotTable view](#)

 [Keys for working with the field list](#)

CTRL+L

Display the field list, or activate it if it's already displayed

Arrow keys

Move to the next item in the field list

SHIFT+UP ARROW

Move to the previous item and include it in the selection

SHIFT+DOWN ARROW

Move to the next item and include it in the selection

CTRL+UP ARROW

Move to the previous item, but don't include the item in the selection

CTRL+DOWN ARROW

Move to the next item, but don't include the item in the selection

CTRL+SPACEBAR

Remove the item from the selection, if the item that has focus is included in the selection, and vice versa

PLUS SIGN (numeric keypad)

Expand the current item in the field list to display its contents. Or expand Totals to display the available total fields.

MINUS SIGN (numeric keypad)

Collapse the current item in the field list to hide its contents. Or collapse Totals to hide the available total fields.

The TAB key

Alternately move to the most recently selected item, the **Add to** button, and the list next to the **Add to** button in the field list

ALT+DOWN ARROW

Open the drop-down list next to the **Add to** button in the field list. Use the arrow keys to move to the next item in the list, and then press ENTER to select an item.

ENTER

Add the highlighted field in the field list to the area in the [PivotTable view](#) that's displayed in the **Add to** list

ALT+F4

Close the field list

[Keys for adding fields and totals](#)

CTRL+SHIFT+S

Add a new total field for the selected field in the [PivotTable view](#) by using the **Sum** summary function

CTRL+SHIFT+C

Add a new total field for the selected field in the PivotTable view by using the **Count** summary function

CTRL+SHIFT+M

Add a new total field for the selected field in the PivotTable view by using the **Min** summary function

CTRL+SHIFT+X

Add a new total field for the selected field in the PivotTable view by using the **Max** summary function

CTRL+SHIFT+E

Add a new total field for the selected field in the PivotTable view by using the **Average** summary function

CTRL+SHIFT+D

Add a new total field for the selected field in the PivotTable view by using the **Standard Deviation** summary function

CTRL+SHIFT+T

Add a new total field for the selected field in the PivotTable view by using the **Standard Deviation Population** summary function

CTRL+SHIFT+V

Add a new total field for the selected field in the PivotTable view by using the **Variance** summary function

CTRL+SHIFT+R

Add a new total field for the selected field in the PivotTable view by using the **Variance Population** summary function

CTRL+SHIFT+B

Turn subtotals and grand totals on or off for the selected field in the PivotTable view

CTRL+F

Add a calculated detail field

[Keys for changing the layout](#)

Note The following four shortcuts do not work if you press the keys 1, 2, 3, or 4

from the numeric pad of your keyboard.

CTRL+1

Move the selected field in the [PivotTable view](#) to the row area

CTRL+2

Move the selected field in the PivotTable view to the column area

CTRL+3

Move the selected field in the PivotTable view to the filter area

CTRL+4

Move the selected field in the PivotTable view to the detail area

CTRL+LEFT ARROW

Move the selected row or column field in the PivotTable view to a higher level

CTRL+RIGHT ARROW

Move the selected row or column field in the PivotTable view to a lower level



[Keys for formatting elements in PivotTable view](#)

To use the following shortcuts, first select a detail field or a data cell for a total field.

The first seven keyboard shortcuts change the number format of the selected field.

CTRL+SHIFT+~ (tilde)

Apply the general number format to values in the selected total or detail field

CTRL+SHIFT+\$

Apply the currency format, with two decimal places and negative numbers in parentheses, to values in the selected total or detail field

CTRL+SHIFT+%

Apply the percentage format, with no decimal places, to values in the selected total or detail field

CTRL+SHIFT+^

Apply the exponential number format, with two decimal places, to values in the selected total or detail field

CTRL+SHIFT+#

Apply the date format, with the day, month, and year, to values in the

selected total or detail field

CTRL+SHIFT+@

Apply the time format, with the hour, minute, and AM or PM, to values in the selected total or detail field

CTRL+SHIFT+!

Apply the numeric format, with two decimal places, thousands separator, and a minus sign for negative values, to values in the selected total or detail field

CTRL+B

Make text bold in the selected field of the [PivotTable view](#)

CTRL+U

Make text underlined in the selected field of the PivotTable view

CTRL+I

Make text italic in the selected field of the PivotTable view



[PivotChart view](#)



[Keys for selecting items in a chart](#)

RIGHT ARROW

Select the next item in the chart

LEFT ARROW

Select the previous item in the chart

DOWN ARROW

Select the next group of items

UP ARROW

Select the previous group of items



[Keys for working with properties and options](#)

ALT+ENTER

Display the **Properties** dialog box

ALT+F4

Close the **Properties** dialog box

The TAB key

When the **Properties** dialog box is active, select the next item on the active tab

RIGHT ARROW

When a tab in the **Properties** dialog box is active, select the next tab
LEFT ARROW

When a tab in the **Properties** dialog box is active, select the previous tab
DOWN ARROW

Display a list or palette when a button that contains a list or palette is selected

SHIFT+F10

Display the shortcut menu

Underlined letter

Carry out a command on the shortcut menu

ESC

Close the shortcut menu without carrying out a command



[Keys for working with fields](#)

ALT+DOWN ARROW

Open the list for the currently selected field

The TAB key

In the drop-down list for a field, alternately move to the most recently selected item, the **OK** button, and the **Cancel** button

Arrow keys

In the drop-down list for a field, move to the next item

SPACEBAR

In the drop-down list for a field, select or clear the check box for the current item

ENTER

Close the drop-down list for a field and apply any changes you made

ESC

Close the drop-down list for a field without applying your changes



[Keys for working with the field list](#)

CTRL+L

Display the field list, or activate it if it's already displayed

Arrow keys

Move to the next item in the field list

SHIFT+UP ARROW

Move to the previous item and include it in the selection

SHIFT+DOWN ARROW

Move to the next item and include it in the selection

CTRL+UP ARROW

Move to the previous item, but don't include the item in the selection

CTRL+DOWN ARROW

Move to the next item, but don't include the item in the selection

CTRL+SPACEBAR

Remove the item from the selection if the item that has focus is included in the selection, and vice versa

PLUS SIGN (numeric keypad)

Expand the current item in the field list to display its contents, or expand Totals to display the available total fields

MINUS SIGN (numeric keypad)

Collapse the current item in the field list to hide its contents, or collapse Totals to hide the available total fields.

The TAB key

In the field list, alternately move to the most recently selected item, the **Add to** button, and the list next to the **Add to** button

ALT+DOWN ARROW

Open the drop-down list next to the **Add to** button in the field list. Use the arrow keys to move to the next item in the list, and then press ENTER to select an item.

ENTER

Add the highlighted field in the field list to the drop area that's displayed in the **Add to** list

ALT+F4

Close the field list

[Show All](#)

What's installed with Access 2003

Some of the content in this topic may not be applicable to some languages.

This topic provides reference information about which features are installed with Microsoft Office Access 2003 and where you can find these features in the installation program. These features are also installed by the Microsoft Office 2003 installation program; they include Microsoft Access features as well as Office Shared Features and Office Tools. The default installation features are installed unless you customize the installation by including or excluding features.



[Installed \(default\)](#)

Feature	Location in the installation program
AutoShapes and Themes	Office Shared Features; Clip Organizer; Clip Organizer Collections
Calendar Control	Microsoft Access for Windows
Document Update Utility	Office Tools
Encapsulated Postscript	Office Shared Features; Converters and Filters; Graphic Filters
GIF (Graphic Interchange Format)	Office Shared Features; Converters and Filters; Graphic Filters
Help for Microsoft Access	Microsoft Access for Windows
Help for Microsoft Graph	Office Tools; Microsoft Graph
Hosted Webs	Office Tools; Office Server Extensions Support
JPEG Interchange Format	Office Shared Features; Converters and Filters; Graphic Filters
Macintosh PICT Import	Office Shared Features; Converters and Filters; Graphic Filters
Microsoft Access Program Files	Microsoft Access for Windows
Microsoft Graph	Office Tools
Microsoft Office Download	Office Shared Features

Control	
Office 2000 Web Components	Office Shared Features
Office 2003 Web Components	Office Shared Features
Office SOAP Toolkit	Office Shared Features
Popular Clip Art	Office Shared Features; Clip Organizer; Clip Organizer Collections
Research Explorer Bar	Office Tools
Save My Settings Wizard	Office Tools
Smart Tag Plug-ins	Office Tools
Spelling and Grammar Checkers	Office Shared Features; Proofing Tools; English
System Information	Office Tools
Tag Image File Format Import	Office Shared Features; Converters and Filters; Graphic Filters
Typical Themes	Office Shared Features; Themes
Typical Wizards	Microsoft Access for Windows
Web Discussions	Office Tools; Office Server Extensions Support
Windows Bitmap Import	Office Shared Features; Converters and Filters; Graphic Filters
Windows Metafile Import	Office Shared Features; Converters and Filters; Graphic Filters
Windows SharePoint Services Support	Office Tools

[Installed on first use \(default\)](#)

Feature	Location in the installation program
Access .NET Programmability Support	Microsoft Access for Windows; .NET Programmability Support
Additional Themes	Office Shared Features; Themes
Additional Wizards	Microsoft Access for Windows
Computer Graphics Metafile Import	Office Shared Features; Converters and Filters; Graphic Filters

Corel Draw Import	Office Shared Features; Converters and Filters; Graphic Filters
Database Replication	Microsoft Access for Windows
Digital Signature for Visual Basic for Applications (VBA) Projects	Office Shared Features
Equation Editor	Office Tools
French Proofing Tools	Office Shared Features; Proofing Tools
Japanese Font	Office Shared Features; International Support
Language Settings Tool	Office Tools
Microsoft Graph .NET Programmability Support	Office Tools; Microsoft Graph; .NET Programmability Support
Microsoft Handwriting Component	Office Shared Features
Microsoft Query	Office Tools
Northwind Database	Microsoft Access for Windows; Sample Databases
Northwind Sample Access Project	Microsoft Access for Windows; Sample Databases
Office 2003 Web Components .NET Programmability Support	Office Shared Features; Office 2003 Web Components; .NET Programmability Support
Office Assistant (six additional Assistants)	Office Shared Features; Office Assistant
PNG File Format Import	Office Shared Features; Converters and Filters; Graphic Filters
Snapshot Viewer	Microsoft Access for Windows
Spanish Proofing Tools	Office Shared Features; Proofing Tools
Speech	Office Shared Features; Alternative User Input
Visual Basic Help	Office Shared Features; Visual Basic for Applications
Web Scripting (Microsoft Script Editor)	Office Tools; HTML Source Editing
	Office Shared Features; Converters and

WordPerfect Graphics Filters Filters; Graphic Filters

[Not installed \(custom installation only\)](#)

Feature	Location in the installation program
Handwriting	Office Shared Features; Alternative User Input
New and Open Office Document Shortcuts	Office Shared Features
Syriac Font	Office Shared Features; International Support
Universal Font	Office Shared Features; International Support

[Other features such as the sample databases and proofing tools](#)

After you have installed Microsoft Access 2003, the sample databases appear on the **Help** menu, under **Sample Databases**, if they were installed.

The Microsoft Access 2003 Setup program may also install Text Converters and additional Proofing Tools; most of these components are installed on first use and therefore do not consume a large amount of disk space. Although you can use these components with other Microsoft Office 2003 applications such as Microsoft Word and Microsoft Excel, they are not designed to be used directly with Microsoft Access.

[Other programs in Microsoft Office 2003](#)

The following programs are installed with all configurations of Microsoft Office 2003, if a typical installation is selected. Additional programs may be installed with other configurations.

Program and location in the Custom dialog box of Setup	Installed with Typical Setup?
Microsoft Excel	Yes
Microsoft Word	Yes

Microsoft PowerPoint
Microsoft Outlook

Yes
Yes

[Show All](#)

Which wizards are installed in Access 2003

Some of the content in this topic may not be applicable to some languages.

 [Wizards installed when you select the "Typical Wizards" component in Setup](#)

Note You can install any of the wizards in this topic by double-clicking **Add/Remove Programs** in Windows Control Panel and running Microsoft Office 2003 Setup.

AutoDialer	Link HTML
AutoForm	Link Spreadsheet
AutoFormat	Link Text
AutoPage	List Box
AutoReport	Lookup
Combo Box	Microsoft Word Mail Merge
Command Button	Page
Crosstab Query	Page Combo Box
Database	Page Command Button
Export	Page List Box
Field Builder	Picture Builder
Form	PivotTable
Import Exchange/Outlook	Print Relationships
Import HTML	Report
Import Spreadsheet	Simple Query
Import Text	Switchboard Manager
Label	Table
Link Exchange/Outlook	

 [Wizards installed when you select the "Additional Wizards" component in](#)

[Setup](#)

Add-in Manager	Microsoft SQL Server Database
Database Splitter	ODBC Connect String Builder
Documenter	Option Group
Find Duplicates Query	Partial Replica
Find Unmatched Query	Performance Analyzer
Graph (Chart)	Security
Input Mask	Subform/Subreport
Link Table Wizard	Table Analyzer
Linked Table Manager	Subform/Subreport Field Linker
Macro To Module Converter	Upsizing

Note The following wizards are always installed: Color Builder, Expression Builder, Query Builder.

[Show All](#)

Install or uninstall menu add-ins

You can install or uninstall [add-ins](#) by using the **Add-in Manager**.

1. On the **Tools** menu, point to **Add-ins**, and then click **Add-in Manager**.

Once the **Add-in Manager** dialog box is displayed, the **Available Add-ins** list displays the add-ins that are currently available. An add-in is already installed if there is an X next to its name. If an add-in is uninstalled, Microsoft Access removes the X but keeps the name in the list.

2. Do one of the following:
 - To add an add-in to the list, click **Add New**, and then specify the add-in's location.
 - To install a currently available add-in, click the add-in name in the **Available Add-ins** list, and then click **Install**.
 - To uninstall a currently available add-in, click the add-in name in the **Available Add-ins** list, and then click **Uninstall**.
3. Click **Close** when you've finished.

Note You should uninstall an add-in before opening it as a database to modify its design.

[Show All](#)

Wizards, builders, and add-ins in Access 2003

[Wizards](#)

Wizard	Description
AutoDialer Wizard	Allows you to specify modem information when you click an AutoDialer button in Form view.
AutoForm Wizard	Creates a form automatically.
AutoFormat Wizard	Applies a predefined format to a form or report.
AutoPage Wizard	Creates a data access page automatically.
AutoReport Wizard	Creates a report automatically.
Chart Wizard	Adds a chart to a form or report based on the data in a table or query.
Combo Box Wizard	Creates a combo box control on a form.
Command Button Wizard	Creates a command button control on a form.
Conflict Resolver Wizard	Resolves conflicts between replicated databases at synchronization time.
Crosstab Query Wizard	Creates a query that summarizes data in a compact, spreadsheet-like format.
Database Splitter Wizard	Splits databases into data and interface portions, so that one or more users can have local copies of the interface connected to the data on a server.
Database Wizard	Creates an entirely new database for a variety of uses.
Documenter	Generates an Access report that displays the design characteristics of database objects .
Export Text Wizard	Exports data to a text file.

Find Duplicates Query Wizard	Creates a query that finds records that have duplicate field values and are in a single table or query.
Find Unmatched Query Wizard	Creates a query that finds records in one table that have no related records in another table.
Form Wizard	Creates a new form.
Import Exchange/Outlook Wizard	Imports an Exchange or Outlook folder to a table in a Microsoft Access database.
Import HTML Wizard	Imports HTML tables and lists from the Internet or an intranet into a Microsoft Access table.
Import Spreadsheet Wizard	Imports a Microsoft Excel or other spreadsheet into a Microsoft Access table.
Import Text Wizard	Imports a text file into a Microsoft Access table.
Input Mask Wizard	Creates an input mask for a field that you choose in a table.
Label Wizard	Creates mailing labels in standard and custom sizes.
Link Exchange/Outlook Wizard	Links an Exchange or Outlook folder to a table in a Microsoft Access database.
Link HTML Wizard	Links an HTML table or list on the Internet or an intranet to a Microsoft Access table.
Link Spreadsheet Wizard	Links spreadsheet data to a Microsoft Access table.
Link Table Wizard	Links to tables in an Access project
Link Text Wizard	Links a text file to a Microsoft Access table.
Linked Table Manager	Manages attachments to tables between databases.
List Box Wizard	Creates a list box control on a form.
Lookup Wizard	Creates a lookup column in a table, which displays a list of values the user can choose from.
Macro To Module Converter	Converts macros to Visual Basic code.
Microsoft Access Mail Merge Wizard	Manages mail merge operations for merging tables, views, functions, or stored procedure into Microsoft Word documents.

Microsoft SQL Server Database Wizard	Creates a new Microsoft SQL Server Database that a new Microsoft Access project is connected to.
Microsoft Word Mail Merge Wizard	Manages mail merge operations by using letters that are stored in Microsoft Word and addresses that are stored in Microsoft Access.
Option Group Wizard	Creates a group of option buttons on a form.
Page Combo Box Wizard	Creates a drop-down control on a data access page.
Page Command Button Wizard	Creates a command button control on a data access page.
Page List Box Wizard	Creates a list box control on a data access page.
Page Wizard	Creates a new data access page.
Partial Replica Wizard	Creates or modifies a partial replica of a replica database.
Performance Analyzer	Analyzes the efficiency of a database and produces a list of suggestions for improving its performance.
PivotTable Wizard	Places a Microsoft Excel PivotTable on a Microsoft Access form.
Print Relationships Wizard	Creates a report that diagrams the relationships in a Microsoft Access database.
Report Wizard	Creates a report that is based on a table or query.
Simple Query Wizard	Creates a select query from the fields that you pick.
Subform/Subreport Field Linker	Links fields in a main form and a subform , or in a main report and a subreport .
Subform/Subreport Wizard	Creates a new subform or subreport on a form or report.
Switchboard Manager	Creates and manages switchboard forms for applications.
Table Analyzer Wizard	Takes a table with a lot of duplicate data and splits it into related tables for more efficient storage.
Table Wizard	Creates a new table.

Upsizing Wizard	Upsizes a Microsoft Access database to a Microsoft SQL Server database.
User-Level Security Wizard	Creates a new, encoded database, with regulated user access, from an existing database. Regulates user access to the current database and creates an unsecured backup copy of the database.

[Builders](#)

Builder	Description
Color Builder	Displays a palette for creating customized colors.
Expression Builder	Creates expressions for macros, queries, and property sheets.
Field Builder	Creates fields in tables.
ODBC Connection String Builder	Creates the correct syntax for a connection to an ODBC database.
Picture Builder	Creates bitmap images for forms and reports.
Query Builder	Creates the correct syntax for a query.
Smart Tags Builder	Displays a list of available smart tags and their actions.

[Add-ins](#)

Add-in	Description
Add-in Manager	Installs and uninstalls wizards, builders, and add-ins.

[Show All](#)

Register an ActiveX control (OLE custom control)

Use this procedure to register an [ActiveX control](#) that is not included with Microsoft Access.

1. Add the files for the control to your hard disk. For more information, see the documentation that came with the control.
2. On the Access **Tools** menu, click **ActiveX Controls**.
3. In the **ActiveX Controls** dialog box, click **Register**.
4. In the **Add ActiveX Control** dialog box, specify a path to the control, and then click **OK**.
5. Click **Close**.

Security Use caution when you are adding ActiveX controls to your Web page. ActiveX controls may be designed in such a way that their use could pose a security risk. We recommend that you use controls from trusted sources only.

Customize the Access environment by setting defaults

1. On the **Tools** menu, click **Options**.
2. Click the tab for the options you want to set.
3. Select the options you want to use.

[Show All](#)

Set command-line options for starting Access

 [Start Microsoft Access with command-line options from the Microsoft Windows Start menu](#)

1. Click the Windows **Start** button, point to **Settings**, and then click **Taskbar & Start Menu**.
2. Click the **Advanced** tab, and then click **Add**.
3. In the **Create Shortcut** dialog box, specify a path to the Microsoft Access program file, or click **Browse**. It's located by default in C:\Program Files\Microsoft Office\Office11.
4. Click to the right of the Access startup command in the box where you type the location of the item, and then type the desired command-line options.

For example, the following command line starts Microsoft Access and opens the Northwind sample database for [exclusive](#) access.

```
"C:\Program Files\Microsoft Office\Office11\MSAccess.exe"  
"C:\Program Files\Microsoft  
Office\Office11\Samples\Northwind.mdb" /excl
```

Search Windows Help for "adding shortcuts" for more information.

 [Start Microsoft Access with command-line options using a shortcut](#)

1. Open the folder where Microsoft Access is installed (located by default in C:\Program Files\Microsoft Office\Office11).
2. Right-click the Microsoft Access program icon, and then click **Create Shortcut**.
3. Right-click the shortcut icon just created, click **Properties**, and then click the **Shortcut** tab.
4. Click to the right of the Microsoft Access startup command in the **Target** box, and then type the desired command-line options.

For example, the following command line starts Microsoft Access and opens the Northwind sample database for [exclusive](#) access.

```
"C:\Program Files\Microsoft Office\Office11\MSAccess.exe"  
"C:\Program Files\Microsoft  
Office\Office11\Samples\Northwind.mdb" /excl
```

Note After you create a shortcut you can put it on your desktop or in a folder, or add it to the **Start** menu. Search Microsoft Windows Help for "adding shortcuts" for more information.

You can also create a shortcut to open any Microsoft Access [database object](#) directly from Windows.

[Show All](#)

Startup command-line options

The following table lists the Microsoft Access command-line options.

Option	Effect
database	Opens the specified Microsoft Access database or Microsoft Access project . Include a path if necessary.
/excl	Opens the specified Access database for exclusive access. To open the database for shared access in a multiuser environment, omit this option. Applies to Access databases only.
/ro	Opens the specified Access database or Access project for read-only access.
/user <i>user name</i>	Starts Access by using the specified user name. Applies to Access databases only.
/pwd <i>password</i>	Starts Access by using the specified password. Applies to Access databases only.
/profile <i>user profile</i>	Starts Access by using the options in the specified user profile instead of the standard Windows Registry settings created when you installed Microsoft Access. This replaces the /ini option used in versions of Microsoft Access prior to Access 97 to specify an initialization file.
/compact <i>target database or target Access project</i>	Compacts and repairs the Access database, or compacts the Access project that was specified before the /compact option, and then closes Access. If you omit a target file name following the /compact option, the file is compacted to the original name and folder. To compact to a different name, specify a target file. If you don't include a path in <i>target database</i> or <i>target Access project</i> , the target file is created in your My Documents folder by default.
/repair	In an Access project, this option compacts the Access project (.adp) file but not the Microsoft SQL Server database. Repairs the Access database that was specified before the /repair option, and then closes Microsoft Access. In Microsoft Access 2000 or later, compact and repair functionality is combined under

/compact. The **/repair** option is supported for backward compatibility.

/convert
target
database

Converts a previous-version Access database or Access project to Access 2000 file format, renames the new file, and then closes Access. You must specify the source database before you use the **/convert** option.

Starts Access and runs the specified macro. Another way to run a macro when you open a database is to use an AutoExec macro.

/x macro

Caution Macros can contain viruses, so you must be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; use digital signatures; maintain a list of trusted sources of macros.

/cmd

Specifies that what follows on the command line is the value that will be returned by the **Command** function. This option must be the last option on the command line. You can use a semicolon (;) as an alternative to **/cmd**.

Use this option to specify a command-line argument that can be used in [Visual Basic](#) code.

/nostartup

Starts Access without displaying the [task pane](#) (the second dialog box that you see when you start Access).

/wrkgrp

workgroup information Starts Access by using the specified workgroup information file.
file Applies to Access databases only.

Notes

- To run a Visual Basic for Applications procedure when you open a database, use the RunCode action in the AutoExec macro or in the macro that you run by using the command-line option **/x**. You can also run a Visual Basic procedure when you open a database by creating a form with a Visual Basic procedure defined for its OnOpen event. Designate this form as the startup form by right-clicking the Database window, clicking

- Startup**, and then entering that form in the **Display Form/Page** box.
- To specify a forward slash (/) or semicolon (;) on the command line, type the character twice. For example, to specify the password **;mjs/md** on the command line, type **;;mjs//md** following the **/pwd** command-line option.

[Show All](#)

Show or hide the task pane at Access startup

The task pane that appears when you start Microsoft Access has options for creating or opening a [Microsoft Access database](#) or a [Microsoft Access project](#). You can show or hide the task pane at Access startup by doing the following:

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **Startup Task Pane** check box.

Show or hide the status bar

Show or hide the status bar for all your Microsoft Access files

1. On the **Tools** menu, click **Options**.
2. On the **View** tab under **Show**, select or clear the **Status bar** check box.

Show or hide the status bar for a specific Access file

1. On the **Tools** menu, click **Startup**.
2. Select or clear the **Display Status Bar** check box.

Note Changes to this setting in the **Startup** dialog box won't take effect until the next time you open the Access file.

[Show All](#)

Change the default folder for new Access databases, Access projects, or pages

1. On the **Tools** menu, click **Options**.
2. Do one of the following:

[Change the default folder for new Microsoft Access databases or Microsoft Access projects](#)

1. Click the **General** tab.
2. In the **Default database folder** box, type the path name of any valid folder where you want to store new [Microsoft Access database](#) files or [Microsoft Access project](#) files.

[Change the default folder for new data access pages](#)

1. Click the **Pages** tab.
2. Under **Default Database/Project Properties**, select the **Use Default Page Folder** check box, and in the **Default Page Folder** box, type the path name of the folder where you want to save new pages.

[Show All](#)

Change the default behavior for selecting or opening database objects

By default, you select a [database object](#) in the [Database window](#) by clicking it, and you open an object by double-clicking it. However, you can change this default behavior so that you open an object by single-clicking it and select it by resting the pointer on it.

1. On the **Tools** menu, click **Options**.
2. On the **View** tab, click **Single-click open**.



You can also select a database object in the Database window by typing the name of the object.

Change default behavior for find or replace operations

1. On the **Tools** menu, click **Options**.
2. Click the **Edit/Find** tab.
3. Under **Default find/replace behavior**, do one of the following:

Select **Fast Search** to search the current field and match the whole field.

Select **General Search** to search all fields and match any part of the field.

Select **Start Of Field Search** to search the current field and match the beginning characters of the field.

Note Changes made to this setting don't take effect until the next time you start Microsoft Access.

[Show All](#)

Set OLE/DDE preferences

1. On the **Tools** menu, click **Options**, and then click the **Advanced** tab.
2. Under **DDE operations**, do one or more of the following:

To ignore [DDE](#) requests from other applications, select **Ignore DDE requests**.

To update DDE links at the interval specified in the **Refresh interval** box, select **Enable DDE refresh**.

3. In the **OLE/DDE timeout** box, enter the number of seconds (0 to 300) Microsoft Access should wait before retrying a failed [OLE](#) or DDE operation.

[Show All](#)

Specify how the insertion point behaves

1. On the **Tools** menu, click **Options**, and then click the **Keyboard** tab.
2. Do one of the following:

[Specify whether you move to the next character or the next field when you press an arrow key](#)

- Under **Arrow key behavior**, do one of the following:

To move the insertion point to the next or previous field after you press the RIGHT or LEFT ARROW key, click **Next field**.

To move the insertion point to the next or previous character after you press the RIGHT or LEFT ARROW key, click **Next character**.

Note If you select **Next field** under **Arrow key behavior**, you can still move the insertion point within a field by pressing F2.

[Prevent the insertion point from moving to another record when you press the RIGHT or LEFT ARROW key](#)

- Do one of the following:

To lock the cursor within the current record, select the **Cursor stops at first/last field** check box. This prevents the insertion point from moving from the first or last field to the previous or next record when you press the RIGHT or LEFT ARROW keys.

To allow the insertion point to move from record to record, clear the **Cursor stops at first/last field** check box.

[Specify how the insertion point behaves when you move to another field](#)

You can specify the location of the insertion point when pressing the ENTER, TAB, or arrow keys to move from field to field on forms and datasheets.

1. Under **Arrow key behavior**, click **Next field**.
2. Under **Behavior entering field**, do one of the following:

To select the entire field when you enter it, click **Select entire field**.

To place the insertion point at the start of the field when you enter it, click **Go to start of field**.

To place the insertion point at the end of the field when you enter it, click **Go to end of field**.

[Specify how the insertion point behaves when you press ENTER](#)

- Under **Move after enter**, do one of the following:

To keep the insertion point in the current field after you press ENTER, click **Don't move**.

To move the insertion point to the next field after you press ENTER, click **Next field**.

To move the insertion point to the current field of the next record after you press ENTER, click **Next record**.

Note These settings do not apply to [data access pages](#).

[Show All](#)

Turn confirmation messages on or off

1. On the **Tools** menu, click **Options**.
2. Click the **Edit/Find** tab.
3. Under **Confirm**, do one or more of the following:

To display a confirmation message when you change a record, select **Record changes**.

To display a confirmation message when you delete a database object, select **Document deletions**.

To display a confirmation message when you run an [action query](#) (Access database only), select **Action queries**.

Note The **Record changes** and **Action queries** options don't apply to [data access pages](#).

Specify the default language sort order (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab.
3. To specify the default language sort order for a new database, click the language that you want to use in the **New database sort order** box.

Note To reset the sort order for an existing database to the language that you selected in the **General** tab, compact that database.

[Show All](#)

Display four-digit years in all fields and controls containing dates

You can ensure that Date fields and [controls](#) display four-digit years, even if the **Format** property settings for individual fields and controls specify dates with two-digit years.

1. On the **Tools** menu, click **Options**, and then click the **General** tab.
2. Under **Use four-digit-year formatting**, do one of the following:

To display four-digit years in the current Microsoft Access file, select the **This database** check box.

To display four-digit years in all Access files, select the **All databases** check box.

Note Selecting one of these check boxes only changes how dates are displayed — it does not change the data itself.

[Show All](#)

Create your own wizards, builders, and add-ins

To develop your own [add-in](#) ([wizard](#), [builder](#), or menu add-in), you can set several database properties and create a USysRegInfo table in the add-in database (*.mda, *.mde) so that it can be installed by using the Add-in Manager. You can also develop a COM add-in that can be installed by the Add-in Manager. For more information, see "Add-ins, Templates, Wizards, and Libraries," in *The Microsoft Office XP Developer's Guide*. For more information about creating a USysRegInfo table, see the [Microsoft Developer Network](#) Web site.

[Show All](#)

Set Access run-time options

Caution Incorrectly editing the registry may severely damage your operating system, requiring you to reinstall it. Microsoft cannot guarantee that problems resulting from editing the registry incorrectly can be resolved. Before editing the registry, back up any valuable data. For the most recent information about using and protecting your computer's registry, see Microsoft Windows Help.

1. Create a Microsoft Access user profile.
2. Under your Access user profile subkey, create a subkey called \Run-Time Options.
3. Do one or more of the following:

[Replace the Access title bar with an application-specific title](#)

1. Create a subkey TitleBar under the \Run Time Options subkey.
2. Set the value of TitleBar as the text you want displayed in the title bar of your application's main window.

Notes

- This setting corresponds to the **Application Title** startup property.
- AppTitleBar is also a valid subkey that Access provides for backward compatibility.

[Replace the Access icon with an application-specific icon](#)

1. Create a subkey Icon under the \Run Time Options subkey.
2. Set the value of Icon as the path to the icon displayed in your application's title bar and when your application is minimized.

Note This setting corresponds to the **Application Icon** startup property.

[Specify an application-specific Help file](#)

1. Create a subkey AppHelpFile under the \Run Time Options subkey.
2. Set the value of AppHelpFile as the path to the default Help file used when a user first gets Help.

 [Display a splash screen that uniquely identifies and announces your application when it starts](#)

1. Create a subkey StartupScreen under the \Run Time Options subkey.
2. Set the value of StartupScreen as the name of the splash screen.

Notes

- A splash screen is different from a startup form and displays before any startup form you might have specified in the **Startup** dialog box. This run-time option takes precedence over a splash screen defined without using a run-time option.
- AppSplashScreen is also a valid subkey that Access provides for backward compatibility.

 [Create a splash screen without using a run-time option](#)

1. Create a [bitmap](#) with the same name as your application and a .bmp extension. For example, if your application name is Taxes.mdb, your bitmap file name would be Taxes.bmp.
2. Put it in the same folder as your application.
3. Using Microsoft Windows Explorer, create a shortcut for your application that specifies the bitmap name on the command line, and then enter it in the **Target** box (**Shortcut Properties** dialog box, **Shortcut** tab). For example:
"C:\Program Files\Microsoft Office\Office 11\MSAccess.exe C:\My Files\Taxes.mdb"

Access looks for such a file when the application opens, and if it finds one, uses it as a splash screen.

Change the font in SQL and query Design View (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab.
3. In the **Query design font** group, select the font and font size.

Note These font settings apply to Access as a whole and are not specific to the Access database or query currently in use.

Set the number of files shown in the list of recently used files

When you start Microsoft Access, when you click **New** on the **File** menu, or when you click **File** on the menu bar, a list of the most recently used files appears. You can click a file name in this list to open the file with the same option settings it had the last time you opened it. You can also change the number of file names that appear on the list of recently used files.

1. On the **Tools** menu, click **Options**.
2. On the **General** tab, do one of the following:
 - To prevent any files from appearing in the list, clear the **Recently used file list** check box.
 - To change the number of files that appear in the list, make sure the **Recently used file list** check box is selected, and then, in the box to its right, select the number of files you want in the list. By default, four files appear in the recently used file list, but you can set the list to contain as many as nine files.

Tips for improving the performance of Microsoft Access and your system

The following guidelines can help you optimize the performance of Microsoft Access on your computer:

- If you're using databases that other users don't need to share, install Microsoft Access and all your databases on your hard disk drive rather than on a network server.
- If you are the only person using a database, open the database for exclusive use; in the **Open** dialog box, click the arrow next to **Open** button, and then click **Open Exclusive**.
- Make more memory available by closing applications that you aren't using.
- Increase RAM on your computer. 40 MB of memory is recommended— 32 MB of memory plus an additional 8 MB of memory for Microsoft Access.
- Don't use any of your RAM for a RAM disk.
- Periodically delete unnecessary files and empty your Recycle Bin, compact your databases, and then defragment your hard disk with the Microsoft Windows Disk Defragmenter. To run the Disk Defragmenter, click the Windows **Start** button, point to **Programs**, point to **Accessories**, point to **System Tools**, and then click **Disk Defragmenter**.
- In most cases, the default virtual memory setting used by Windows should perform optimally. However, in some situations, adjusting virtual memory parameters can improve performance. If you've already tried deleting unnecessary files and you still have a performance problem, try changing the default virtual memory setting in the following cases:
 - You don't have much disk space available on the drive that is currently being used for virtual memory, and another local drive with space is available.
 - Another local drive is available that is faster than the current drive (unless that disk is heavily used).

In these cases, you might get better performance by specifying a different drive for virtual memory.

You also might get better performance by specifying that the disk space

available for virtual memory be at least 25 MB minus available RAM. For example, if your computer has 12 MB of RAM, you should specify at least 13 MB of virtual memory. You may want to specify more if you are running several large applications.

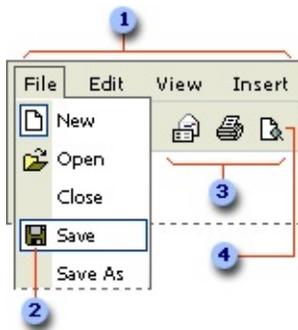
To change Windows virtual memory parameters, in Windows Control Panel, double-click the System icon, click **Performance Options** under the **Advanced** tab, click the **Change** button in the Virtual Memory section, and then specify a different hard disk, or enter a value in the **Initial size** box that is at least 25 minus your available RAM.

- If you have a wallpaper (full-screen background) bitmap on your Windows desktop, replace it with a solid color or pattern bitmap, or no bitmap at all.
- If you use a screen saver, use a blank screen saver or consider not using one at all.
- Eliminate disk compression software, or consider moving your databases to an uncompressed drive.
- To ensure optimal performance, use substitution fonts only if you need dual-font support to display all of your data.

[Show All](#)

About menus and toolbars

A [menu](#) displays a list of commands. Some of these commands have images next to them so you can quickly associate the command with the image. Most menus are located on the [menu bar](#), which is the [toolbar](#) at the top of the screen. Toolbars can contain buttons, menus, or a combination of both.



1 Menu bar

2 Menu command

3 Toolbar

4 Button

[Displaying only the commands and buttons that you use](#)

Microsoft Office automatically customizes menus and toolbars based on how often you use the commands. When you first start an Office program, only the most basic commands appear. Then, as you work, the menus and toolbars adjust so that only the commands and toolbar buttons you use most often appear.

[Displaying all the commands on a menu](#)

To look for a command that you don't use often or have never used before, click the arrows  at the bottom of the menu to show all the commands. You can also double-click the menu to expand it. When you expand one menu, all of the menus are expanded until you choose a command or perform another action.

When you click a command on the expanded menu, the command is immediately added to the short version of the menu. If you do not use the command often, it is dropped from the short version of the menu.

[Positioning toolbars on the same row](#)

Toolbars can be positioned next to each other in the same row. When you put multiple toolbars in the same row, there might not be enough room to display all of the buttons. If there isn't enough room, the buttons that you have used most recently are displayed.

[Seeing all of the toolbar buttons](#)

You can resize to display more buttons, or you can show all buttons on a toolbar. To see a list of buttons that won't fit on a built-in [docked toolbar](#), click **Toolbar Options**  at the end of the toolbar. When you use a button that is not displayed on the toolbar, that button is moved to the toolbar, and a button that has not been used recently is dropped to the Toolbar Options list.

[Customizing menus and toolbars](#)

You can customize menus and toolbars yourself; you can add and remove buttons and menus on toolbars, create your own custom toolbars, hide or display toolbars, and move toolbars. You can customize the menu bar the same way you customize any [built-in toolbar](#)— for example, you can quickly add and remove buttons and menus on the menu bar— but you can't hide the menu bar.

[Customizing your applications](#)

Using Microsoft Access, you can create your own applications with a user interface that has the "look and feel" of a Microsoft Windows application. In addition to the toolbar features common to all Office applications, you can do the following:

- Create, delete, and rename your own custom toolbars, menu bars (including submenus), and [shortcut menus](#).
- Convert the custom toolbars from previous-version macros.
- Import toolbars from another [Access data file](#).

- Enhance custom toolbars by adding keys and help information to buttons and commands.
- Protect changes to toolbars, such as movement and resizing.
- Attach toolbars to a form or report.
- Add Microsoft Visual Basic [functions](#) to buttons and commands.

Note In the [PivotTable list](#), [spreadsheet](#), and [chart](#) tools on a [data access page](#), certain toolbar functionalities differ from the rest of Microsoft Access.

[Show All](#)

Designate a global menu bar or shortcut menu

By using your own [global menu bar](#), you control what tasks users can perform in your [Microsoft Access file](#) by limiting and customizing the menus and commands that are available. The global menu bar replaces the built-in menu bar for all Microsoft Access windows, except where you've added a custom menu bar for a form or report. You also can define a [global shortcut menu](#) to replace built-in shortcut menus for [datasheets](#), forms, form [controls](#), and reports.

1. On the **Tools** menu, click **Startup**.
2. Do one or more of the following:

To set a global menu bar, in the **Menu Bar** box, click the name of a custom menu bar from the current Access database or Access project, or click **(default)** to display the built-in menu bar.

To set a global shortcut menu bar, in the **Shortcut Menu Bar** box, click the name of a custom shortcut menu from the current Access database or Access project, or click **(default)** to use the built-in shortcut menus.

3. To prevent users from switching to the built-in menu bar from the custom menu bar by pressing CTRL+F11, clear the **Use Access Special Keys** check box.

Notes

- Changes to this setting in the **Startup** dialog box won't take effect until the next time the Access file is opened.
- You can designate or change the global menu bar or shortcut menu in a Microsoft Visual Basic for Applications procedure by setting the **MenuBar** property or **ShortcutMenuBar** property of the **Application** object.

[Show All](#)

Create a toolbar

[Create a toolbar for all Microsoft Access data files](#)

To design a toolbar that's available in all your [Microsoft Access databases](#) and [Microsoft Access projects](#), add buttons to the blank [built-in toolbars](#), **Utility 1** and **Utility 2**.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, select the check box for **Utility 1** or **Utility 2** in the **Toolbars** box.
3. Close the **Customize** dialog box. Access displays the toolbar you selected in step 2.
4. To complete the toolbar, add buttons from the **Customize** dialog box, move or copy a button from another toolbar, or add menus.

[Create a custom toolbar](#)

1. On the **Tools** menu, click **Customize**.
2. Click the **Toolbars** tab.
3. Click **New**.
4. In the **Toolbar name** box, type the name you want, and then click **OK**.
5. Click the **Commands** tab.
6. Do one of the following:

[Add a button to the toolbar](#)

1. Click a category in the **Categories** box.
2. Drag the command you want from the **Commands** box to the displayed toolbar.

[Add a built-in menu to the toolbar](#)

1. In the **Categories** box, click **Built-in Menus**.
2. Drag the menu you want from the **Commands** box to the displayed toolbar.

7. When you have added all the buttons and menus you want, click **Close**.

Note Custom [toolbars](#) that you create are specific to the [Access data file](#). If you want to use a custom toolbar in a different Access data file, you must re-create or import the custom toolbar.

[Create a toolbar from a macro](#)

1. In the Database window, click **Macros**  under **Objects**, and then click the name of the macro you want to create a toolbar from.
2. On the **Tools** menu, point to **Macro**, and then click **Create Toolbar from Macro**.

Note Microsoft Access won't delete the original macro (or macros if you're working with a menu macro and the associated menu macro groups) after it creates the new toolbar. The new toolbar still depends on the original macro or macros, except for macros that contain only AddMenu actions or RunCommand (formerly DoMenuItem) actions.

[Create a custom menu bar for the current Access data file](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **New**.
3. In the **Toolbar Name** box, type the name you want, and then click **OK**.

The new menu bar is now placed next to the **Customize** dialog box.

4. On the **Toolbars** tab, click **Properties**.
5. In the **Type** list, click **Menu Bar**.
6. Set any other properties you want, and then click **Close**.
7. To complete the menu bar, add custom menus or built-in menus.

Note You can attach the custom menu bar to a form or report or you can designate the bar as the [global menu bar](#).

[Create a custom shortcut menu](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **New**.

3. In the **Toolbar Name** box, type the name you want, and then click **OK**.
4. On the **Toolbars** tab, click **Properties**.
5. In the **Type** list, click **Popup**.
6. Set the **AllowCustomizing** property the way you want, and then click **Close**.

Microsoft Access adds the shortcut menu to the **Shortcut Menus** toolbar, from which you customize all shortcut menus.

7. In the **Toolbars** box on the **Toolbars** tab, click **Shortcut Menus**.
8. On the **Shortcut Menus** toolbar, click the **Custom** category.
9. To complete the menu, add commands from the **Customize** dialog box and move or copy commands from other menus.

Note You can attach the custom shortcut menu to a form, report, or control, or designate the menu as the [global shortcut menu](#).

[Create a storage toolbar](#)

When you delete a [built-in menu](#), the menu remains in the **Customize** dialog box. However, when you delete a custom menu, it is permanently deleted. To remove a custom menu and save it for later use, create a storage [toolbar](#) for storing your unused custom menus.

1. Show the toolbar that contains the custom button you want to store.

[How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - If the toolbar name is listed, click the toolbar name.
 - If the toolbar name is not listed:
 1. Click **Customize**.
 2. In the **Customize** dialog box, click the **Toolbars** Tab, and then double-click the toolbar you want to show.
2. On the **Tools** menu, click **Customize**, and then click the **Toolbars** tab.
3. Click **New**.
4. In the **Toolbar name** box, type the name you want for the storage toolbar.
5. Click **Close**.

The new storage toolbar remains showing on the screen.

6. Do one of the following for each custom toolbar button you want to store:
 - To move a button, hold down ALT and drag the button to the new storage toolbar.
 - To copy a button, hold down CTRL+ALT and drag the button to the new storage toolbar.
7. To hide the storage toolbar, right-click the storage toolbar and clear the check box next to its name on the [shortcut menu](#).

[Show All](#)

Delete a toolbar, button, or command

[Delete a custom toolbar](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Toolbars** tab.
3. In the **Toolbars** box, select the custom [toolbar](#) you want to delete.
4. Click **Delete**.

Note You can't delete a [built-in toolbar](#). When you select a built-in toolbar, the **Reset** button appears, which when clicked restores the toolbar to its default buttons, menus, and [submenus](#).

[Delete a menu](#)

1. Make sure the [toolbar](#) you want to change is visible.

[How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - Click the toolbar you want to display.
 - To view more toolbars, click **Customize**. In the **Customize** dialog box, click the **Toolbars** tab, and then select the check box for the toolbar you want to display. Click **Close**.
2. On the **Tools** menu, click **Customize**.
3. With the **Customize** dialog box open, drag the [menu](#) you want to delete off the toolbar.

[Tip](#)

When you delete a [built-in menu](#), the menu is still in the **Customize** dialog box. However, when you delete a custom menu, it is permanently deleted. To remove a custom menu and save it for later use, create a storage toolbar for storing unused menus, move the menu to this storage toolbar, and then hide the storage toolbar.

[Delete a custom shortcut menu](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, click the name of the custom shortcut menu you want to delete.
4. In the **Type** list, change the type to **Toolbar**, and then close the property sheet.

The name of the shortcut menu now appears in the **Toolbars** box on the **Toolbars** tab.

5. Select the name of your custom shortcut menu in the **Toolbars** box, and then click **Delete**.

[Delete a toolbar button](#)

1. Make sure the [toolbar](#) you want to change is visible.

[How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - Click the toolbar you want to display.
 - To view more toolbars, click **Customize**. In the **Customize** dialog box, click the **Toolbars** tab, and then select the check box for the toolbar you want to display. Click **Close**.
2. On the toolbar, click **Toolbar Options** .
3. Point to **Add or Remove Buttons**, and do one of the following:

[To delete a built-in button from a toolbar or built-in floating toolbar](#)

1. Point to the toolbar name.
2. Clear the check box next to the button you want to delete.

[To delete a button from a custom toolbar](#)

1. Click **Customize**.

2. With the **Customize** dialog box open, on the toolbar, right-click the button you want to delete.
3. Click **Delete**.
4. On the **Customize** dialog box, click **Close**.

Note To delete a button that is not available in the Add or Remove Buttons list, hold down ALT and then drag the button off the toolbar.

[Delete a menu command](#)

1. On the **Tools** menu, click **Customize**.
2. In the program window, click the [menu](#) that contains the command you want to delete.
3. Drag the command you want to delete off the menu.
4. On the **Customize** dialog box, click **Close**.

[Show All](#)

Rename a toolbar, command, or button

[Rename a custom toolbar](#)

You can rename any custom [toolbar](#) that you have created; however, you cannot rename a [built-in toolbar](#) that comes with Microsoft Office.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Toolbars** tab.
3. In the **Toolbars** box, click the custom toolbar you want to rename.
4. Click **Rename**.
5. Type a new name in the **Toolbar name** box, and then press ENTER.

[Rename a menu](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. With the **Customize** dialog box open, right-click the menu you want to rename, type the new name in the **Name** box on the shortcut menu, and then press ENTER.
3. On the **Customize** dialog box, click Close.

[Rename a custom shortcut menu](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, click the shortcut menu you want to rename.
4. In the **Toolbar Name** box, type the new name.

[Rename a menu command or toolbar button](#)

1. Make sure the [toolbar](#) you want to change is visible.

[How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - Click the toolbar you want to display.
 - To view more toolbars, click **Customize**. In the **Customize** dialog box, click the **Toolbars** tab, and then select the check box for the toolbar you want to display. Click **Close**.
2. On the **Tools** menu, click **Customize**.
3. Do one of the following:
 - To rename a [menu](#) command, click the menu that contains the command, right-click the command, type a name in the **Name** box, and then press ENTER.
 - To rename a toolbar button, right-click the button, type the name in the **Name** box, and then press ENTER.

Note If the toolbar button doesn't display text, you won't see the name change except when you view the [ScreenTip](#) by resting the mouse pointer on the button.

Add a submenu to a menu

You can add a submenu to a built-in or custom menu.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If the toolbar, menu bar, or shortcut menu isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar, menu bar, or shortcut menu name you want to show.
3. In the **Customize** dialog box, click the **Commands** tab.
4. In the **Categories** box, click **New Menu**.
5. From the **Commands** box, drag **New Menu** over the menu you want to add the submenu to until the top-level menu drops down, and then add **New Menu** in the desired location.
6. Right-click the new submenu, and then type a name in the **Name** box on the shortcut menu. Press ENTER.
7. Complete the menu by adding commands from the **Customize** dialog box, or by moving or copying commands from other menus.

[Show All](#)

Customize a shortcut menu

1. On the **Tools** menu, click **Customize**.
2. Click the **Toolbars** tab.
3. Under **Toolbars**, select the **Shortcut Menus** check box.
4. Do one of the following:

[Delete a command from a shortcut menu](#)

1. Click a category on the **Shortcut Menu toolbar**, and then click the [shortcut menu](#) that contains the command.
2. Drag the command off the menu.

[Add a command to a shortcut menu](#)

1. Click a category on the **Shortcut Menu toolbar**, and then click the [shortcut menu](#) you want to modify.
2. In the **Customize** dialog box, click the **Commands** tab.
3. In the **Categories** box, click a category for the command.
4. Drag the command from the **Commands** box to where you want it to appear on the shortcut menu, and then release the mouse button.

[Show All](#)

Add a function to a button or command

 [Add a custom button that runs a Microsoft Visual Basic function to a toolbar](#)

1. Create the function you want the button to run.

 [How?](#)

1. To open a [module](#), do one of the following:

To open a new [standard module](#), in the Database window, click **Modules**  under **Objects**, and then click the **New** button on the Database window toolbar.

To open an existing standard module, click **Modules**  under **Objects**, select the module you want to open, and then click **Design**.

To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the toolbar.

To open a new [class module](#) that isn't associated with a form or report, in the Database window click **Class Module** on the **Insert** menu.

To open an existing class module, in the Database window, click **Modules**  under **Objects**, select the module you want to open, and then click the **Design** button on the Database window toolbar.

2. Declare the function by typing the **Function** statement.
3. Type a function name immediately followed by any function [arguments](#) in parentheses. For example, the following declaration for the **IsLoaded** function specifies *strFormName* as an argument:

```
Function IsLoaded (strFormName As String) As Boolean
```

4. Add the [Microsoft Visual Basic](#) code that performs the operation or calculation that you want the function to perform.
2. On the **View** menu, point to **Toolbars**, and then click **Customize**.
3. If the toolbar isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar name you want to show.
4. In the **Customize** dialog box, click the **Commands** tab.
5. In the **Categories** box, click **File**.
6. Drag the **Custom** command from the **Commands** box to the desired toolbar within the program window.
7. With the **Customize** dialog box open, right-click the button on the toolbar. On the shortcut menu, type a new name for the button in the **Name** box.
8. On the same shortcut menu, click **Properties**.
9. In the **On Action** box, type the name of the function you want to run in the format `=functionname()`. For example, for a custom function named `SetCaption()`, type `=SetCaption()`. For a built-in function, enter the name of the function and any required arguments; for example `=MsgBox(IIf(Instr(Time(), "PM"), "Good Afternoon", "Good Morning"))`.
10. Specify any other properties for the button.

[Add a custom command that runs a Visual Basic function to a menu](#)

1. Create the function you want the command to run.

[How?](#)

1. To open a [module](#), do one of the following:

To open a new [standard module](#), in the Database window, click **Modules**  under **Objects**, and then click the **New** button on the Database window toolbar.

To open an existing standard module, click **Modules**  under **Objects**, select the module you want to open, and then click **Design**.

To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the toolbar.

To open a new [class module](#) that isn't associated with a form or report,

in the Database window click **Class Module** on the **Insert** menu.

To open an existing class module, in the Database window, click **Modules**  under **Objects**, select the module you want to open, and then click the **Design** button on the Database window toolbar.

2. Declare the function by typing the **Function** statement.
3. Type a function name immediately followed by any function [arguments](#) in parentheses. For example, the following declaration for the **IsLoaded** function specifies *strFormName* as an argument:

```
Function IsLoaded (strFormName As String) As Boolean
```

4. Add the [Microsoft Visual Basic](#) code that performs the operation or calculation that you want the function to perform.
2. On the **View** menu, point to **Toolbars**, and then click **Customize**.
3. If the toolbar, menu bar, or shortcut menu isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar, menu bar, or shortcut menu name you want to show.
4. In the **Customize** dialog box, click the **Commands** tab.
5. In the **Categories** box, click **File**.
6. Drag the **Custom** command from the **Commands** box over the menu on the menu bar or the toolbar, or under the appropriate category on the **Shortcut Menus** toolbar. When the menu displays a list of menu commands (or an empty box if it's new), point to the location where you want the command to appear on the menu, and then release the mouse button.
7. With the **Customize** dialog box open, right-click the command on the menu. On the shortcut menu, type a new name for the command in the **Name** box.
8. On the same shortcut menu, click **Properties**.
9. In the **On Action** box, type the name of the function you want to run in the format *=functionname()*. For example, for a custom function named `SetCaption()`, type **=SetCaption()**. For a built-in function, enter the name of the function and any required arguments; for example **=MsgBox(IIf(Instr(Time()), "PM"), "Good Afternoon", "Good Morning"))**.
10. Specify any other properties for the command.

Note If you add a command to a built-in menu (for example, the **Edit** menu),

that command will appear in all views containing that built-in menu.

[Show All](#)

Add an access key to a menu, command, or button

[Assign an access key to a menu or command](#)

You can use an [access key](#) instead of the mouse to move the [focus](#) to a menu or command. For example, press the access keys ALT+T to move the focus to the **Tools** menu, then press the access key O to select the **Options** command.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If the toolbar, menu bar, or shortcut menu isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar, menu bar, or shortcut menu name you want to show.
3. With the **Customize** dialog box open, right-click the menu or command in the program window you want to specify an access key for.
4. In the **Name** box, type an ampersand (&) before the character you want to use as the access key.

On the menu or command, Microsoft Access adds an underscore to the character you've designated as the access key.

Note Use a different access key for each command on a given menu.

[Create shortcut key text for a built-in toolbar button or menu command](#)

This procedure only creates the [shortcut key](#) text. It doesn't assign the action you want the key to perform.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If the toolbar isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar name you want to show.
3. With the **Customize** dialog box open, right-click the [built-in toolbar](#) button or menu command in the program window you want to specify shortcut text for.
4. Click **Properties** on the shortcut menu.

5. In the **Shortcut Text** property box, type the shortcut key or key combination that you want to display. (The **Shortcut Text** property isn't available for custom buttons or commands.)
6. To assign the action you want the shortcut keys to perform, create an AutoKeys macro.

 [How?](#)

You can assign an [action](#) or set of actions to a specific key or key combination by creating an AutoKeys [macro group](#). When you press the key or key combination, Microsoft Access carries out the action. If you assign an action to a key combination that is already being used by Access (for example, CTRL+C is the key combination for **Copy**), the action you assign this key combination replaces the Access key assignment.

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click **New**.
3. Click **Macro Names**  on the toolbar.
4. In the **Macro Name** column, type the key or key combination to which you want to assign the action or set of actions

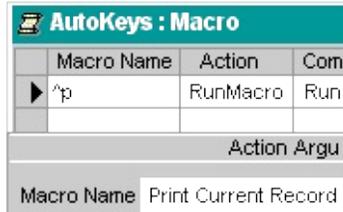
Syntax for AutoKeys key combinations

The following table shows the key combinations you can use to make key assignments in an AutoKeys [macro group](#). These key combinations are a subset of the syntax used in the **SendKeys** statement in [Microsoft Visual Basic](#).

SendKeys syntax	Key combination
^A or ^4	CTRL+Any letter or number key
{F1}	Any function key
^{F1}	CTRL+Any function key
+{F1}	SHIFT+Any function key
{INSERT}	INS
^{INSERT}	CTRL+INS
+{INSERT}	SHIFT+INS
{DELETE} or {DEL}	DEL

^{DELETE} or ^{DEL} CTRL+DEL
+{DELETE} or +{DEL} SHIFT+DEL

5. Add the action or set of actions you want the key or key combination to carry out. For example, you could add a RunMacro action that runs the Print Current Record macro when CTRL+P is pressed.



6. Repeat steps 4 and 5 for any other key assignments you want to make.
7. Save the macro group with the name AutoKeys.

The new key assignments are in effect as soon as you save the macro group and each time you open the database.

Note For a command on a menu, the shortcut key text appears to the right of the command. For a toolbar button, the shortcut key text appears in the [ToolTip](#), and only if you have selected the **Show shortcut keys In ScreenTips** box on the **Options** tab within the **Customize** dialog box. This dialog box is accessed through the **View** menu, by pointing to **Toolbars** and then clicking on **Customize**.

Convert a previous-version menu macro to a menu

This procedure creates the new style of menu bar or shortcut menu from macros you created in a previous version of Microsoft Access.

1. In the Database window, under **Objects**, click **Macros**  and then click the name of the macro you want to create a menu bar or shortcut menu for.

Select only the top-level menu bar macro. You don't need to select the macro group for each menu that appears on the menu bar.

2. On the **Tools** menu, point to **Macro**, and then click **Create Menu from Macro** or **Create Shortcut Menu from Macro**.

Note Access won't delete the original macro (or macros if you're working with a menu bar macro and the associated menu macro groups) after it creates the new menu bar or shortcut menu. The new menu bar still depends on the original macro or macros, except for macros that contain only AddMenu actions or RunCommand (formerly DoMenuItem) actions.

[Show All](#)

Import menus and toolbars

You can import custom menus and toolbars from a [Microsoft Access database](#) or [Microsoft Access project](#). However, Access won't import a toolbar, menu bar, or shortcut menu if it has the same name as one in the [Access data file](#) you're importing to— you must rename it before importing.

1. Open a database.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. In the **Files of type** box, make sure **Microsoft Access (*.mdb, *.adp)** is selected.
4. In the **Look in** box, select the drive and folder for the Microsoft Access database (.mdb or .adp) that you want to import from, and then double-click the database.
5. In the **Import Objects** dialog box, click **Options**.
6. Under **Import**, click **Menus and Toolbars**.

[Show All](#)

Restore original toolbar, button, and command settings

Do one or more of the following:

[Reset all default settings for a built-in toolbar or menu bar to display either the original buttons or the menus and commands and to restore their original property settings.](#)

Microsoft Access will also restore the original property settings for the toolbar or menu bar itself, including restoring its screen location and size, and hiding or showing the bar.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to reset default settings for.
4. Click the **Restore Defaults** button.

The **Restore Defaults** button isn't available until Microsoft Access detects that the selected [built-in toolbar](#) or menu bar has been modified.

[Restore a menu's original appearance](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. With the **Customize** dialog box open, right-click the menu you want to restore, and then click **Reset** on the shortcut menu.
3. On the **Customize** dialog box, click **Close**.

Note The **Reset** command restores the built-in menu to its original images, commands, and submenus.

[Restore original buttons, menus, and commands on a built-in toolbar, menu bar, or shortcut menu without changing other properties of the toolbar or menu bar \(such as the screen location and size\).](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Toolbars** tab.
3. In the **Toolbars** box, click the name of the toolbar you want to restore.
4. Click **Reset**.

Notes

- You cannot reset a custom toolbar.
- As you work with Microsoft Office, the menus and toolbars are personalized to show the commands that you use most often if you have cleared the **Always show full menus** box on the **Options** tab in the **Customize** dialog box. If you want, you can return the toolbar and menu display to the state it was in when you first started using Office.

[Restore how an individual toolbar button or menu command originally appeared \(except in an individual shortcut menu\)](#)

1. Make sure the toolbar you want to change is visible.

[How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - If the toolbar name is listed, click the toolbar name.
 - If the toolbar name is not listed:
 1. Click **Customize**.
 2. In the **Customize** dialog box, click the **Toolbars** Tab, and then double-click the toolbar you want to show.
2. On the **View** menu, point to **Toolbars**, and then click **Customize**.
3. With the **Customize** dialog box open, do one of the following:
 - To restore a toolbar button, right-click the button, and then click **Reset** on the shortcut menu.
 - To restore a menu command, click the menu that contains the command, right-click the command, and then click **Reset** on the shortcut menu.

Notes

- The Reset command restores the built-in toolbar button or menu command to its original image, name, and command.
- The Reset command isn't available for a button that displays a list when clicked.



[Show default toolbar buttons and menu commands](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Options** tab.
3. Click **Reset my usage data**.

Notes

- The **Reset my usage data** button affects the buttons shown on a [built-in toolbar](#) only if it is not wide enough to display all the buttons. It affects the menu commands shown on the short version of built-in menus only if you have the **Always show full menus** box cleared on the **Options** tab in the **Customize** dialog box.
- The **Reset my usage data** button affects only the toolbar buttons and menu commands of the Microsoft Office program in which the button is clicked.
- The **Reset my usage data** button does not change the location of toolbars, does not remove any buttons or commands you've added by using the **Customize** dialog box, and does not add buttons or commands you've deleted.

[Show All](#)

Create a custom toolbar

1. On the **Tools** menu, click **Customize**.
2. Click the **Toolbars** tab.
3. Click **New**.
4. In the **Toolbar name** box, type the name you want, and then click **OK**.
5. Click the **Commands** tab.
6. Do one of the following:

[Add a button to the toolbar](#)

1. Click a category in the **Categories** box.
2. Drag the command you want from the **Commands** box to the displayed toolbar.

[Add a built-in menu to the toolbar](#)

1. In the **Categories** box, click **Built-in Menus**.
 2. Drag the menu you want from the **Commands** box to the displayed toolbar.
7. When you have added all the buttons and menus you want, click **Close**.

Note Custom [toolbars](#) that you create are specific to the [Access data file](#). If you want to use a custom toolbar in a different Access data file, you must re-create or import the custom toolbar.

[Show All](#)

Attach a menu or toolbar to a form or report

 [Attach a custom menu bar to a form or report](#)

Once attached, the custom menu bar replaces the built-in menu bar for the form or report.

1. Create the menu bar.

 [How?](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **New**.
3. In the **Toolbar Name** box, type the name you want, and then click **OK**.

The new menu bar is now placed next to the **Customize** dialog box.

4. On the **Toolbars** tab, click **Properties**.
5. In the **Type** list, click **Menu Bar**.
6. Set any other properties you want, and then click **Close**.
7. To complete the menu bar, add custom menus or built-in menus.

Note You can attach the custom menu bar to a form or report or you can designate the bar as the [global menu bar](#).

2. Open the form or report in Design view.
3. Click the [form selector](#) or [report selector](#).
4. On the **Form Design** or **Report Design** toolbar, click **Properties** .
5. In the **MenuBar** property box, specify the menu bar name.

Repeat this procedure for each form or report you want to attach the menu bar to.

Repeat this procedure for each form or report you want to attach the menu bar to.

[Attach a custom toolbar to a form or report](#)

Once attached, the toolbar replaces the [built-in toolbar](#) for the form or report.

1. Create the toolbar.

[How?](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **New**.
3. In the **Toolbar name** box, type the name you want, and then click **OK**.
4. Click the **Commands** tab.
5. Do one of the following:

[Add a button to the toolbar](#)

1. Click a category in the **Categories** box.
2. Drag the command you want from the **Commands** box to the displayed toolbar.

[Add a built-in menu to the toolbar](#)

1. In the **Categories** box, click **Built-in Menus**.
 2. Drag the menu you want from the **Commands** box to the displayed toolbar.
6. When you have added all the buttons and menus you want, click **Close**.

Note Custom [toolbars](#) that you create are specific to the [Access data file](#). If you want to use a custom toolbar in a different Access data file, you must re-create or import the custom toolbar.

2. Open the form or report in Design view.
3. Click the [form selector](#) or [report selector](#) in Design view.
4. On the toolbar, click **Properties** .
5. In the **Toolbar** property box, specify the toolbar name.

Repeat this procedure for each form or report you want to attach the toolbar to.

Repeat this procedure for each form or report you want to attach the toolbar to.

[Attach a shortcut menu to a form, form control, or report](#)

Once attached, the shortcut menu replaces the built-in shortcut menu for a form, a [control](#), or a report.

1. Create the shortcut menu.

[How?](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **New**.
3. In the **Toolbar Name** box, type the name you want, and then click **OK**.
4. On the **Toolbars** tab, click **Properties**.
5. In the **Type** list, click **Popup**.
6. Set the **AllowCustomizing** property the way you want, and then click **Close**.

Microsoft Access adds the shortcut menu to the **Shortcut Menus** toolbar, from which you customize all shortcut menus.

7. In the **Toolbars** box on the **Toolbars** tab, click **Shortcut Menus**.
8. On the **Shortcut Menus** toolbar, click the **Custom** category.
9. To complete the menu, add commands from the **Customize** dialog box and move or copy commands from other menus.

Note You can attach the custom shortcut menu to a form, report, or control, or designate the menu as the [global shortcut menu](#).

2. Open the form or report in Design view.
3. Click the [form selector](#), [report selector](#), or control in Design view.
4. On the toolbar, click **Properties** .
5. In the **ShortcutMenuBar** property box, specify the shortcut menu name.

For a form, also make sure the **ShortcutMenu** property is set to **Yes**.

Repeat this procedure for each form, report, or control you want to attach

the shortcut menu to.

Repeat this procedure for each form, report, or control you want to attach the shortcut menu to.

[Show All](#)

Display palettes permanently while designing a form or report

To keep a color palette or style palette displayed the entire time you're designing a form, report, or data access page, drag the palette away from its toolbar button.

1. In form [Design view](#), report Design view, or data access page Design view, click the arrow next to the **Fill/Back Color** , **Font/Fore Color** , **Line/Border Color** , **Line/Border Width** , or **Special Effect**  buttons on the **Formatting** toolbar.
2. Click the bar at the top of the palette and drag the palette away from the button.

[Show All](#)

Add a button to dial a selected phone number

You must have a working Hayes or Hayes-compatible modem to use this procedure.

1. Open a table or form in [Datasheet view](#) or a form in [Form view](#).
2. On the **View** menu, point to **Toolbars**, and then click **Customize**.
3. Click the **Commands** tab in the **Customize** dialog box.
4. Click **Records** in the **Categories** list.
5. Drag **AutoDialer**  to the Form view or Datasheet view toolbar, or to a custom toolbar.
6. Click **Close**.
7. In the form or datasheet, select the phone number you want to dial.
8. Click **AutoDialer**  on the toolbar.
9. Check to see that the phone number is entered correctly, and then click **OK**.

Note You can change your modem options by clicking **Setup** in the **AutoDialer** dialog box.

[Show All](#)

Control the display of all built-in toolbars and menus

You can control whether built-in toolbars, the built-in menu bar, or built-in shortcut menus are available in a [Microsoft Access file](#).

1. On the **Tools** menu, click **Startup**.
2. Do one or more of the following:

To prevent or allow access to all built-in toolbars, clear or select the **Allow Built-in Toolbars** check box.

To prevent or allow access to built-in shortcut menus, clear or select the **Allow Default Shortcut Menus** check box.

To prevent or allow switching between the custom menu bar and the built-in menu bar by pressing CTRL+F11, clear or select the **Use Access Special Keys** check box.

Notes

- Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.
- Clearing or selecting the **Allow Built-in Toolbars** check box does not affect toolbars for PivotTable lists or spreadsheets on a [data access page](#).

[Show All](#)

Control whether some or all of the built-in menus are displayed

If the [Microsoft Access file](#) you're creating provides the user access to the built-in menus, you can choose to display only a predefined subset of the menus. That way, you can hide menus, such as **View** and **Insert**, that would enable a user to access the [Design view](#) of objects in your Access database or Access project and to change their design.

1. On the **Tools** menu, click **Startup**.
2. Clear or select the **Allow Full Menus** check box.

Note Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.

[Show All](#)

Prevent or allow the changing of toolbars, buttons, or commands

[Prevent or allow changes to specific toolbars, buttons, or commands](#)

[Prevent or allow adding, moving, or removing specific toolbars, menu bars, or shortcut menus, and setting properties for buttons, commands, or menus](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar, menu bar, or custom shortcut menu you want to set properties for.
4. Select or clear the **Allow Customizing** box.

[Prevent or allow movement of specific toolbars or menu bars](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to set properties for.
4. Select or clear the **Allow Moving** check box.

Note This setting is not affected by the **Allow Toolbar/Menu Changes** setting in the **Startup** dialog box (**Tools** menu).

[Prevent or allow resizing of specific floating toolbars or menu bars](#)

You can resize [floating](#) toolbars or menu bars.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to set properties for.
4. Select or clear the **Allow Resizing** check box.

Note This setting is not affected by the **Allow Toolbar/Menu Changes** setting in the **Startup** dialog box (**Tools** menu).

[Prevent or allow showing and hiding specific toolbars or menu bars](#)

By default, you can show and hide custom toolbars and menu bars by choosing them from the submenu you see when you point to **Toolbars** on the **View** menu or from the shortcut menu you get when you right-click on a toolbar button or menu. You can prevent users from doing this by removing the names of the toolbars or menu bars from these menus.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to set properties for.
4. Do one of the following:
 - To show a toolbar, select the **Allow Showing/Hiding box** and **Show On Toolbars Menu** check boxes.
 - To hide a toolbar, clear the **Allow Showing/Hiding box** and **Show On Toolbars Menu** check boxes.

If you clear this option, make sure you clear both options. If you don't clear the **Show On Toolbars Menu** check box, the toolbar or menu bar will appear on the menu, but you won't be able to select it.

[Prevent or allow specific toolbar or menu bar docking](#)

You can prevent or allow vertical, horizontal, or all [docking](#) of toolbars.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to set properties for.
4. In the **Docking** box, click the setting you want to use.

Notes

- For any setting other than **Can't Change**, make sure that the **Allow Moving** check box is also selected.

- This setting is not affected by the **Allow Toolbar/Menu Changes** setting in the **Startup** dialog box (**Tools** menu).

[Prevent or allow changes to some or all toolbars, buttons, or commands](#)

[Prevent or allow changes to all toolbars and command bars](#)

You can control whether users are able to change any built-in or custom toolbars and menu bars in a [Microsoft Access file](#).

1. If you also want to prevent users from moving, sizing, or docking toolbars or menu bars, you must first set the appropriate properties in the **Customize** dialog box for each toolbar or menu bar in your application.
2. On the **Tools** menu, click **Startup**.
3. Clear or select the **Allow Toolbar/Menu Changes** check box.

Notes

- Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.
- This option does not affect toolbars for PivotTable lists or spreadsheets on a [data access page](#).

[Control whether some or all of the built-in menus are displayed](#)

If the [Microsoft Access file](#) you're creating provides the user access to the built-in menus, you can choose to display only a predefined subset of the menus. That way, you can hide menus, such as **View** and **Insert**, that would enable a user to access the [Design view](#) of objects in your Access database or Access project and to change their design.

1. On the **Tools** menu, click **Startup**.
2. Clear or select the **Allow Full Menus** check box.

Note Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.

[Control the display of all built-in toolbars and menus](#)

You can control whether built-in toolbars, the built-in menu bar, or built-in shortcut menus are available in a [Microsoft Access file](#).

1. On the **Tools** menu, click **Startup**.
2. Do one or more of the following:

To prevent or allow access to all built-in toolbars, clear or select the **Allow Built-in Toolbars** check box.

To prevent or allow access to built-in shortcut menus, clear or select the **Allow Default Shortcut Menus** check box.

To prevent or allow switching between the custom menu bar and the built-in menu bar by pressing CTRL+F11, clear or select the **Use Access Special Keys** check box.

Notes

- Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.
- Clearing or selecting the **Allow Built-in Toolbars** check box does not affect toolbars for PivotTable lists or spreadsheets on a [data access page](#).

[Show All](#)

Troubleshoot toolbars

[My built-in toolbars and menus have disappeared.](#)

If all the [built-in toolbars](#) and some of the built-in menus have disappeared from all views, you may need to change the settings in the **Startup** dialog box or temporarily bypass its settings. You should also check the **Allow Showing/Hiding** and **Show On Toolbars Menu** property settings of the individual toolbar in the **Customize** dialog box (**View** menu, **Toolbars** command).

[To change the settings](#)

Click **Startup** on the **Tools** menu, and then click **Allow Built-in Toolbars** to display all the [built-in toolbars](#), and click **Allow Full Menus** to display all the built-in menus. Close the [Microsoft Access data file](#), and then reopen it.

If you're still missing some menus, they may have been removed from the menu bar. You can re-add specific built-in menus to the menu bar.

[To temporarily bypass the settings in the Startup dialog box](#)

You can temporarily bypass all the settings specified in the **Startup** dialog box instead of resetting the options. Hold down the bypass key (the SHIFT key) while you open the database.

[I want to return built-in toolbars that came with Microsoft Access to their original settings.](#)

Do one or more of the following:

[Reset all default settings for a built-in toolbar or menu bar to display either the original buttons or the menus and commands and to restore their original property settings.](#)

Microsoft Access will also restore the original property settings for the toolbar or

menu bar itself, including restoring its screen location and size, and hiding or showing the bar.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. On the **Toolbars** tab, click **Properties**.
3. In the **Selected Toolbar** box, select the name of the toolbar or menu bar you want to reset default settings for.
4. Click the **Restore Defaults** button.

The **Restore Defaults** button isn't available until Microsoft Access detects that the selected [built-in toolbar](#) or menu bar has been modified.

[Restore a menu's original appearance](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. With the **Customize** dialog box open, right-click the menu you want to restore, and then click **Reset** on the shortcut menu.
3. On the **Customize** dialog box, click **Close**.

Note The **Reset** command restores the built-in menu to its original images, commands, and submenus.

[Restore original buttons, menus, and commands on a built-in toolbar, menu bar, or shortcut menu without changing other properties of the toolbar or menu bar \(such as the screen location and size\).](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Toolbars** tab.
3. In the **Toolbars** box, click the name of the toolbar you want to restore.
4. Click **Reset**.

Notes

- You cannot reset a custom toolbar.
- As you work with Microsoft Office, the menus and toolbars are personalized to show the commands that you use most often if you have cleared the **Always show full menus box** on the **Options** tab in the **Customize** dialog box. If you want, you can return the toolbar and menu display to the state it was in when you first started using Office.

 [Restore how an individual toolbar button or menu command originally appeared \(except in an individual shortcut menu\)](#)

1. Make sure the toolbar you want to change is visible.

 [How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - If the toolbar name is listed, click the toolbar name.
 - If the toolbar name is not listed:
 1. Click **Customize**.
 2. In the **Customize** dialog box, click the **Toolbars** tab, and then double-click the toolbar you want to show.
2. On the **View** menu, point to **Toolbars**, and then click **Customize**.
3. With the **Customize** dialog box open, do one of the following:
 - To restore a toolbar button, right-click the button, and then click **Reset** on the shortcut menu.
 - To restore a menu command, click the menu that contains the command, right-click the command, and then click **Reset** on the shortcut menu.

Notes

- The Reset command restores the built-in toolbar button or menu command to its original image, name, and command.
- The Reset command isn't available for a button that displays a list when clicked.

 [Show default toolbar buttons and menu commands](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. Click the **Options** tab.
3. Click **Reset my usage data**.

Notes

- The **Reset my usage data** button affects the buttons shown on a [built-in toolbar](#) only if it is not wide enough to display all the buttons. It affects the

menu commands shown on the short version of built-in menus only if you have the **Always show full menus** box cleared on the **Options** tab in the **Customize** dialog box.

- The **Reset my usage data** button affects only the toolbar buttons and menu commands of the Microsoft Office program in which the button is clicked.
- The **Reset my usage data** button does not change the location of toolbars, does not remove any buttons or commands you've added by using the **Customize** dialog box, and does not add buttons or commands you've deleted.

 [I'm having trouble either pasting or resetting an icon image onto a menu or toolbar.](#)

The command on which the icon appears may be set to a style that doesn't allow images. To complete the operation you were trying to perform, first change the command's style.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If it isn't already displayed, show the toolbar or menu containing the command whose style you want to change.

 [How?](#)

1. On the **View** menu, point to **Toolbars**.
2. Do one of the following:
 - If the toolbar name is listed, click the toolbar name.
 - If the toolbar name is not listed:
 1. Click **Customize**.
 2. In the **Customize** dialog box, click the **Toolbars** tab, and then double-click the toolbar you want to show.
3. Leave the **Customize** dialog box open. On the toolbar or menu, right-click the command whose style you want to change.
4. On the shortcut menu, click **Default Style** or **Image And Text**.

Note On a menu, the **Default Style** setting displays a command's text and associated icon image (if it has one). On a toolbar or menu bar, the **Default Style** setting displays just the icon image.

 [I can't customize a toolbar.](#)

If you are unable to customize a toolbar, make sure that the **Allow Customizing** check box in the **Toolbar Properties** dialog box is selected.

If the **Toolbars** command on the **View** menu is disabled, you might need to change the settings in the **Startup** dialog box or temporarily bypass its settings.

[To change the settings](#)

- On the **Tools** menu, click **Startup**, and select the **Allow Toolbar/Menu Changes** check box. Close the database, and then reopen it.

[To temporarily bypass the settings in the **Startup** dialog box](#)

You can temporarily bypass all the settings specified in the **Startup** dialog box instead of resetting the options. Hold down the bypass key (the SHIFT key) while you open the database.

[I can't see my entire toolbar on the screen.](#)

Some buttons may not appear on the toolbar because there is not enough room to display them. Click **More Buttons**  at the end of the toolbar to see a list of additional buttons.

There may not be enough room to display all of the buttons because the toolbar is on the same row as another toolbar. To see more buttons, you can resize the toolbar or move the toolbar to its own row or to another location. To move the toolbar, click on the handle at the left end of the toolbar and drag the toolbar to the desired location on the screen.

[I can't show or hide a toolbar, menu bar, or shortcut menu.](#)

If you are unable to show or hide a toolbar, menu bar, or shortcut menu, make sure that the **Allow Showing/Hiding** check box in the **Toolbar Properties** dialog box is selected.

[I can't find a menu command, toolbar button, or dialog box option.](#)

The menu might not be expanded If arrows  appear at the bottom of the [menu](#), the menu item might be available on the expanded menu. Click the arrows, and then click the command you want. You can also double-click the menu to expand it.

There might not be enough room to display all the buttons If the toolbar is on the same row as another toolbar, there might not be enough room to display all the buttons. Click **Toolbar Options** , and then click the button you want.

You might have a different language setting The command or control doesn't appear if you don't have editing enabled for the language that the command or control applies to. You need to enable editing for the language you want to work with.

The Microsoft Office program you're using may not be maximized Some toolbar buttons may be hidden if your program window is not maximized. Click the **Maximize**  button to enlarge the program window to its fullest extent.

 [I created a custom toolbar and shared it with another person, but it looks different on their computer.](#)

The buttons and commands that you share with another person are always available, but whether or not they appear on the short version of the [menu](#), or on the [toolbar](#) when there isn't enough room to display all the buttons, depends on the individual menu and toolbar settings for that person.

Microsoft Office stores the commands and buttons you've used frequently and recently as your personal menu and toolbar settings. When the **Always show full menus** check box is cleared (**Toolbar Options** arrow , **Add or Remove buttons, Customize** dialog box, **Options** tab), these personalized settings affect the commands that appear on each menu. Additionally, if there isn't enough room on a toolbar to display all the buttons (such as when you position a toolbar on the same row as another toolbar), personalized settings affect which buttons do appear.

[Show All](#)

About startup options

You can use startup options to control how a [Microsoft Access file](#) looks and behaves when it opens. The startup options apply only to the current Access file.

The startup properties don't override the property settings for a specific toolbar, menu bar, shortcut menu, form, or report. For example, the **Menu Bar** and **Shortcut Menu Bar** options in the **Startup** dialog box don't override the property settings for specific forms or reports that have a custom menu bar or [shortcut menu](#). Thus, when you open a form or report, Microsoft Access displays the custom menu bar attached to the form or report instead of the [global menu bar](#) specified in the **Startup** dialog box.

You can also create an AutoExec macro to carry out an action whenever an Access file opens. You can use the **Startup** dialog box instead of or in addition to an AutoExec macro. An AutoExec macro runs after the startup options have taken effect; therefore, you should avoid any actions in an AutoExec macro that change the effect of the startup option settings. For example, if you specify a form in the **Display Form/Page** box in the **Startup** dialog box, and you also use the **OpenForm** action in an AutoExec macro, Microsoft Access first displays the form specified in the **Startup** dialog box, then immediately displays the form specified in the **OpenForm** action.

Caution Macros can contain viruses, so you must be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; use digital signatures; maintain a list of trusted sources of macros.

In a Microsoft Access database, if [user-level security](#) has been defined for your database, you can prevent users from changing the settings in the **Startup** dialog box by making sure they haven't been granted Administer permission for the database.

[Show All](#)

Ignore startup options

If you used the **Startup** dialog box (**Tools** menu) or created an AutoExec macro or an Open [event procedure](#) to specify what happens when you open a Microsoft Access application, you can bypass those settings to regain full access to your Access file.

- Hold down the Bypass key (the SHIFT key) while you open the database.

Depending on the macro security settings, you might see one or more security messages on startup. You must continue to keep the Bypass key pressed until you have closed the security messages.

Note You can disable the bypass key by writing a macro or Microsoft Visual Basic for Applications code that sets the **AllowBypassKey** property of the database to **False**.

[Show All](#)

Control how an Access file looks and behaves when it opens

You can use startup options to specify, for example, what form to display, whether toolbars can be customized, and whether [shortcut menus](#) are available in your [Microsoft Access file](#). You can also use a special [macro](#) named AutoExec to carry out an [action](#) or series of actions when your database first opens. When you open a database, Microsoft Access looks for a macro with this name and, if it finds one, runs it automatically.

Caution Macros can contain viruses, so you must be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; use digital signatures; maintain a list of trusted sources of macros.

Set startup options

1. On the **Tools** menu, click **Startup**.
2. Select the options or enter the settings you want to use.

[Create a macro that runs when an Access file first opens](#)

1. Create a [macro](#) containing the actions you want to run when you open the database.
2. Save the macro with the name AutoExec.

The next time you open the database, Microsoft Access runs this macro automatically.

Note If you don't want to run the AutoExec macro when you open the database, hold down the SHIFT key when the database opens.

[Show All](#)

Display a form or data access page at startup

1. On the **Tools** menu, click **Startup**.
2. In the **Display Form/Page** box, click a form or [data access page](#) from the current [Microsoft Access file](#).

Notes

- Changes to these settings in the **Startup** dialog box won't take effect until the next time the [Microsoft Access database](#) or [Microsoft Access project](#) is opened.
- The Northwind sample database has a startup form. To view this form, open the Northwind database in your Microsoft Office folder's Samples folder. The Startup form appears automatically when you open Northwind, but you can view it whenever you want by clicking **Forms**  under **Objects** in the [Database window](#), clicking **Startup**, and then clicking the **Open** button on the Database window toolbar.

[Show All](#)

Control the display of the Database window

You can change the settings that determine whether the [Database window](#) is displayed when a [Microsoft Access file](#) opens. If you hide the Database window, you need to display a startup form or [data access page](#) so that others can navigate in your application.

1. On the **Tools** menu, click **Startup**.
2. Do one or more of the following:

To display or hide the Database window when a [Microsoft Access database](#) or [Microsoft Access project](#) opens, clear or select the **Display Database Window** check box.

To prevent or allow users from displaying the Database window by pressing F11, clear or select the **Use Access Special Keys** check box.

3. If you have cleared both the **Use Access Special Keys** check box and the **Display Database Window** check box, replace the **File** menu with your own custom menu. This prevents the Database window from appearing when users repeatedly open the same Access file from the list of most recently used files.

Notes

- Changes to these settings in the **Startup** dialog box won't take effect until the next time the Access file is opened.
- The Northwind sample database has a Main Switchboard form— an example of a form you might use to control navigation in your Access file. To view this form, open the Northwind database in your Microsoft Office folder's Samples folder. Click **Forms**  under **Objects** in the Database window, click Main Switchboard, and then click the **Open** button on the Database window toolbar.

Display a specific title in the application window title bar

1. On the **Tools** menu, click **Startup**.
2. In the **Application Title** box, enter the title you want displayed in the application window's title bar.

Note Changes to the **Application Title** setting take effect immediately upon closing the **Startup** dialog box.

Display a custom icon for an application

1. On the **Tools** menu, click **Startup**.
2. To display a custom icon in the title bar of your application, enter the name of the .ico or .bmp file that contains the icon you'd like to use in the **Application Icon** box. If you don't know the name of the file, click the **Build** button next to the box, and then use the **Icon Browser** to locate the file.
3. To also display the icon in the title bar of your forms and reports, select the **Use as Form and Report Icon** check box. A custom icon appears in the title bar of forms and reports in Access 2002 or later. If you open your application in a previous version of Microsoft Access, the icon will appear in the title bar of the application, but not in the title bar of the application's form and reports.

Notes

- Changes to the **Application Icon** setting take effect immediately upon closing the **Startup** dialog box.
- If you'll be distributing your application, it's recommended that the icon file reside in the same folder as your Microsoft Access application.

Turn on or off Windows theming for form controls

In a database created using Office Access 2003, most form controls automatically inherit the Windows theme, if you are using Microsoft Windows XP SP 1, and the chosen theme is not **Windows Classic**. Controls that do not inherit the Windows theme are labels, images, object frames, subforms, lines, and rectangles.

1. On the **Tools** menu, click **Options**.
2. On the **Forms/Reports** tab, select or clear the **Use Windows Themed Controls on Forms** check box.

Note To change the Windows theme, open the **Display** window in Control Panel. On the **Themes** tab, select a theme other than **Windows Classic** in the **Themes** list box.

[Show All](#)

Set ANSI SQL query mode (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Caution Be careful when changing the [ANSI SQL query mode](#). Existing [queries](#) may not run or may return unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.



[Set the ANSI SQL query mode for the current database](#)

1. On the **Tools** menu, click **Options**, and then click the **Tables/Queries** tab.
2. Do one of the following:
 - Select the **This database** check box to set the query mode to ANSI-92 SQL.
 - Clear the **This database** check box to set the query mode to ANSI-89 SQL.

Note Microsoft Access closes, compacts, and then re-opens the database before the new setting takes effect.



[Set the ANSI SQL query mode default for new databases](#)



[ANSI-89](#)

1. On the **Tools** menu, click **Options**, and then click the **Tables/Queries** tab.
2. Clear the **Default for new databases** check box.



[ANSI-92](#)

1. On the **Tools** menu, click **Options**, and then click the **Advanced** tab.
2. Select **Access 2002 - 2003** from the **Default File Format** list box.
3. Click the **Tables/Queries** tab.
4. Select the **Default for new databases** check box.

Notes

- ANSI-89 is the default setting for a new [Microsoft Access database](#) in 2002 - 2003 and 2000 file format. You cannot set the SQL query mode new database default to ANSI-92 in 2000 file format because the option is disabled; ANSI-89 is the only query mode setting available for a database in Access 2000 file format.
- Although you can set the query mode to ANSI-92 in a [replica](#), Access will ignore it. You can only set this option in the [Design Master](#) of a [replica set](#).

[Show All](#)

Change the default data type and text size (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can change the default [data type](#) and the default size of a text field for a new column created in Database Designer or in Table Designer.

1. On the **Tools** menu, click **Options**, and then click the **Tables/Queries** tab.
2. Do one or more of the following:
 - To set the default data type for a column, select a Microsoft SQL server data type from the drop-down list box under **Default field type**.
 - To set the default size for a text field ([char](#), [nchar](#), [varchar \(n\)](#), [nvarchar \(n\)](#)), enter a number in the **Text** text box under **Default field sizes**.

[Show All](#)

Special keys that can be turned on or off at startup

Selecting or clearing the **Use Access Special Keys** check box in the **Startup** dialog box enables or disables all of the following keys in a [Microsoft Access file](#).

Keys	Result
F11	Brings the Database window to the front.
CTRL+G	Brings up the Immediate window.
CTRL+F11	Toggles between the custom menu bar and the built-in menu bar.
CTRL+BREAK	In a Microsoft Access project , stops Access from retrieving records from the server.
ALT+F11	Starts the Microsoft Visual Basic Editor.

[Show All](#)

Prevent users from stopping the retrieval of records from the server (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

In a [Microsoft Access project](#), you can press CTRL+BREAK to stop the retrieval of records from the server, and you can also prevent CTRL+BREAK from stopping the retrieval of records.

1. On the **Tools** menu, click **Startup**.
2. Clear or select the **Use Access Special Keys** check box.

Note Changes to this setting in the **Startup** dialog box won't take effect until the next time the Access project is opened.

[Show All](#)

Prevent or allow access to the Visual Basic Editor or the Immediate window

In a [Microsoft Access file](#), you can prevent the Microsoft Visual Basic Editor from starting when you press ALT+F1 and the Immediate window from showing when you press CTRL+G.

1. On the **Tools** menu, click **Startup**.
2. Clear or select the **Use Access Special Keys** check box.

Note Changes to this setting in the **Startup** dialog box won't take effect until the next time the Access file is opened.

[Show All](#)

About customizing your application's environment with user profiles and run-time options

Caution Incorrectly editing the registry may severely damage your operating system, requiring you to reinstall it. Microsoft cannot guarantee that problems resulting from editing the registry incorrectly can be resolved. Before editing the registry, back up any valuable data. For the most recent information about using and protecting your computer's registry, see Microsoft Windows Help.

You can use Microsoft Access user profiles and run-time options to customize your application's environment and provide an additional level of customization that complements and enhances startup options and command-line options.



[Benefits of using an Access user profile](#)

An Access user profile is a special set of Microsoft Windows registry keys that you can create to override standard Access and [Microsoft Jet database engine](#) settings, and to specify additional run-time options. You then use the **/profile** command-line option to specify this Access user profile when you start your application.

Access provides database properties and startup options to meet most of your customization needs. However, if you're an application developer, you might want to take advantage of the added benefits of an Access user profile to customize your application's environment. For example, you can use an Access user profile to:

- Prevent standard Access settings that users might modify in their database properties and startup options from conflicting with your application needs.
- Ensure that when you distribute your applications, other programs that use Jet do not override custom settings.
- Test your application under different configurations.

Note A user profile that you use to start Access from the command line is not the same thing as a user profile that's defined for logging on to the operating system. An Access user profile applies only to Access, and only when you start Access from the command line. A user profile defined for the operating system applies to every application on the operating system and is used to maintain system data for individual users.

[Benefits of using run-time options](#)

You can add run-time options to your Access user profile to further customize your application's environment. For example, you can:

- Replace the Access title bar with an application-specific title.
- Replace the Access icon with an application-specific icon.
- Specify an application-specific Help file as the default Help file.
- Add a splash screen to uniquely identify and announce your application when it starts.

[Using methods to verify user profiles and to dynamically modify Jet registry values](#)

You can use the **acSysCmdProfile** constant in the *action* argument of the **SysCmd** method to determine whether your application has properly loaded your Access user profile (for example, your users might accidentally open your application through Windows Explorer and bypass the **/profile** command-line option).

You can also use the [Data Access Object \(DAO\) SetOption](#) method of the **DBEngine** object to dynamically modify Jet registry values at run time for a single session, without permanently modifying the values in the Microsoft Windows registry.

[Show All](#)

Create an Access user profile

Caution Incorrectly editing the registry may severely damage your operating system, requiring you to reinstall it. Microsoft cannot guarantee that problems resulting from editing the registry incorrectly can be resolved. Before editing the registry, back up any valuable data. For the most recent information about using and protecting your computer's registry, see Microsoft Windows Help.

1. Start the Registry Editor



1. Click the **Microsoft Windows Start** button.
2. Click **Run**.
3. Type **regedit**.
4. Click **OK**.
2. Navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE` subkey.
3. Add three nested subkeys by using the following format to ensure uniqueness: `\Your Company\Your Application\Your Application Version Number`.
4. Under the `\Your Application Version Number` subkey, re-create any of the subkeys located under the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access` subkey that contain values that you want to modify (except the `\Profiles` subkey, which you will create in step 7), and then copy the appropriate values under each subkey.
5. Modify the values that you want to change.
6. Navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access` subkey.
7. Add a new subkey, and name it **\Profiles**.
8. Add a string value to the `\Profiles` subkey, and give it the name that you want to use on the command line.
9. For the value's data, enter a string that points to the location of the subkeys you created in step 3: `SOFTWARE\Your Company\Your Application\Your Application Version Number`

10. Create a command line for your application that specifies the name you entered in step 8, and enclose the name in double quotation marks ("). For example: `MSAccess.exe /profile "Your Profile" "Your Application.mdb"`

Override any of the Microsoft Jet 4.0 database engine default settings for Access

1. Navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Your Company\Your Application\Your Application Version Number` profile subkey.
2. Create a subkey named `\Jet\4.0\Engines`.
3. For the first default setting you want to override, add an appropriately named subkey below the subkey created in step 2. For example, to override Jet Paradox settings, add a subkey named Paradox.
4. Add values of the appropriate name and type to that subkey, and then specify the settings.

You can model your subkeys and values after the subkeys and values in `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines`.

5. Repeat steps 3 and 4 for each default setting you want to override.

Note In a [Microsoft Access database](#) or [Microsoft Access project](#), it's possible to have the same registry key set in three different ways; but only one value can be used. In a case like this, an Access user profile setting takes precedence over an Access database or Access project setting, which in turn takes precedence over a Microsoft Jet setting.

[Show All](#)

Customize driver settings

Caution Many Microsoft Access registry settings are added automatically when you install Microsoft Access. Changing the installed settings can produce unexpected results, which may make it necessary to reinstall Microsoft Access. It is strongly recommended that you not change settings in the Microsoft Windows registry. However, if you must change registry settings or options, use the Windows Registry Editor, and change only those settings or options that are explicitly documented. The revised settings take effect the next time you start Microsoft Access. It is recommended that you back up all of the registry, or at least the part you will be editing. For information, look up "backing up the registry" in the index of the Registry Editor Help.

Notes

- Microsoft Jet database engine settings apply only to [Microsoft Access databases](#), not to [Microsoft Access projects](#).
- You can also use the [Data Access Object \(DAO\)](#) method, **Application.DBEngine.SetOption**, to dynamically modify Microsoft Jet registry values at run-time for a single session, without permanently modifying the values in the Windows registry.

[Customize text driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc`

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Text` subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access text data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\J` subkey, and then modify values there. If the `\Text` subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize Microsoft Excel driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Excel subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Microsoft Excel data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Office Access 2003, navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\Jet subkey, and then modify values there. If the \Excel subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

 [Customize dBASE driver settings](#)

1. Start the Registry Editor.

 [How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.

2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc`

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Xbas` subkey and make the desired changes.

Important Changes to settings in these subkeys will affect all programs installed on the computer that use the Microsoft Jet database engine to access dBASE data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\J` subkey, and then modify values there. If the `\dBASE` subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

 [Customize Paradox driver settings](#)

1. Start the Registry Editor.

How?

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc`

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Paradox` subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Paradox data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\Jet` subkey, and then modify values there. If the `\Paradox` subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize Lotus driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc`

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Lotu` subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Lotus 1-2-3 data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\J` subkey, and then modify values there. If the `\Lotus` subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize Microsoft Exchange driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc`

`\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Exch` subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Microsoft Exchange data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\J subkey, and then modify values there. If the \Exchange subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize ODBC driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\ODB subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access [ODBC databases](#). In addition to Microsoft Access, this includes

Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Office\\11.0\\Access\\J
subkey, and then modify values there. If the \\ODBC subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize HTML driver settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Office\\11.0\\Acc

\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Jet\\4.0

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Jet\\4.0\\Engines\\Text
subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access HTML data. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the
\\HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Office\\11.0\\Access\\J
subkey, and then modify values there. If the \\Text subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Show All](#)

Customize Jet database settings (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Caution Many Microsoft Access registry settings are added automatically when you install Microsoft Access. Changing the installed settings can produce unexpected results, which may make it necessary to reinstall Microsoft Access. It is strongly recommended that you not change settings in the Microsoft Windows registry. However, if you must change registry settings or options, use the Windows Registry Editor, and change only those settings or options that are explicitly documented. The revised settings take effect the next time you start Microsoft Access. It is recommended that you back up all of the registry, or at least the part you will be editing. For information, look up "backing up the registry" in the index of the Registry Editor Help.

Notes

- Microsoft Jet database engine settings apply only to [Microsoft Access databases](#), not to [Microsoft Access projects](#).
- You can also use the [Data Access Object \(DAO\)](#) method, **Application.DBEngine.SetOption** to dynamically modify Microsoft Jet registry values at run-time for a single session, without permanently modifying the values in the Windows registry.

[Customize Microsoft Jet version 2.x database engine settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys

that contain settings that affect Microsoft Access are:

\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Acc

\HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Jet\4.0

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\4.0\Engines\Jet 2.x subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Microsoft Access 1.x and 2.0 databases. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic.

To make changes that apply only to Microsoft Office Access 2003, navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\Jet\4.0\Engines\Jet 2.x subkey, and then modify values there. If the \Jet 2.x subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

 [Customize Microsoft Jet version 3.x database engine settings](#)

1. Start the Registry Editor.

How?

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

```
\HKEY_LOCAL_MACHINE\SOFTWARE  
\Microsoft\Office\11.0\Access
```

```
\HKEY_LOCAL_MACHINE\SOFTWARE  
\Microsoft\Jet\4.0
```

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the
\HKEY_LOCAL_MACHINE
\SOFTWARE\Microsoft\Jet\4.0\Engines\Jet 3.x subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Microsoft Access 95 or Microsoft Access 97 databases. In addition to Microsoft Access, this includes Microsoft Excel and Microsoft Visual Basic.

To make changes that apply only to Microsoft Office Access 2003, navigate to the
\HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Office\11.0\Access\Jet\4.0\Engines
\Jet 3.x subkey, and then modify values there. If the \Jet 3.x subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Customize Microsoft Jet version 4.0 database engine settings](#)

1. Start the Registry Editor.

[How?](#)

1. To run the Registry Editor, click the Microsoft Windows **Start** button, click **Run**, type **regedit**, and then click **OK**.
2. Navigate to the appropriate registry subkey. The two main subkeys that contain settings that affect Microsoft Access are:

`\HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Office\11.0\Access`

`\HKEY_LOCAL_MACHINE\SOFTWARE
\Microsoft\Jet\4.0`

If there is a plus (+) to the left of the subkey, that registry subkey contains additional subkeys. To edit a registry subkey's value, double-click the subkey in the left pane of the Registry Editor window to open it, and then double-click the value you want to edit in the right pane of the Registry Editor window. In some cases, you may need to add new subkeys or values. For information on how to add new subkeys and values, use the Registry Editor Help.

2. Do one of the following:

To make changes to the [Microsoft Jet database engine](#), navigate to the `\HKEY_LOCAL_MACHINE
\SOFTWARE\Microsoft\Jet\4.0\Engines\Jet 4.0` subkey and make the desired changes.

Important Changes to settings in this subkey will affect all programs installed on the computer that use the Microsoft Jet database engine to access Microsoft Access 2000 and later databases. In addition to Microsoft Access, this includes Microsoft Excel 2000 and later and Microsoft Visual

Basic 6.0 or later.

To make changes that apply only to Microsoft Office Access 2003, navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Office\11.0\Access\Jet\4.0\Engines\Jet 4.0 subkey, and then modify values there. If the \Jet 4.0 subkey does not exist, create the subkey, and then add the values.

3. Exit the Registry Editor.
4. Exit and restart Microsoft Access to use the new settings.

[Show All](#)

About creating custom Help

There are several ways to add custom Help to a Microsoft Access [form](#), [report](#), or [control](#).

Display tips for controls and toolbar buttons

You can display tips that appear over controls when the user moves the mouse pointer over the control. You can also create tips for new toolbar buttons that you create.

Display text in the status bar

You can display helpful information about a form or control in the [status bar](#).

Display custom Help

You can display more extensive Help by creating a Help file and displaying it when the user presses F1 or clicks the **What's This** button  on the Form window.

- For forms and reports, you can display Help when the user presses F1. You can also display Help for a control that has [focus](#).
- For forms and controls, you can display Help when the user clicks the **What's This** button and then clicks the background of the form or a control.

You create the Help file by using a program that can save documents in Rich Text Format, such as Microsoft Help Workshop, or in HTML format, such as Microsoft HTML Help Workshop.

Note You can download Microsoft Help Workshop from the MSDN Web site.

[Show All](#)

Create custom Help

You can display custom help using WinHelp for a form, report, or a control. Your Help topics will be displayed when a user presses F1. Additionally, you can add a **What's This** button to a form.

You can also use Microsoft HTML Help Workshop to display custom help. For more information, see the MSDN Web site.

[Display Help when a user presses F1](#)

1. Create a Help file.

[How?](#)

1. Create the source file by using the text editor or word-processing program of your choice.
 2. Save the source file in Rich Text Format.
 3. Compile the Help file by using Microsoft Help Workshop.
2. Open the form or report in [Design view](#).
 3. Double-click the [form selector](#) or [report selector](#) to display the property sheet.
 4. In the **HelpFile** property box, type the name of the compiled Help file.

Note For best results, put the Help file in the same folder as your application. That way, you can keep the setting for the **HelpFile** property the same for each site that uses your application, but allow users to install the application in the folder of their choice. If you omit a path setting in this property, Microsoft Access looks for the Help file in the folder where your application is installed.

5. In the **HelpContextID** property box for the form or report, type the number of the topic (a number other than 0) that will be displayed when a user presses F1, or clicks the **What's This** button  and then clicks the background of the form. This topic would generally cover assistance for the

whole form or report.

6. In the **HelpContextID** property box for each control on the form that can receive the [focus](#), type the number of the topic that will be displayed when the user either presses F1 when the control has focus, or clicks the **What's This** button  and then clicks the control. You must map the values set for the **HelpContextID** property to the context strings in your Help system. For more information, see the Help file that comes with Microsoft Help Workshop or Microsoft HTML Help Workshop.

Notes

- If a control doesn't have a custom Help topic, the custom Help topic for the form will be displayed when the user either presses F1 when the control has focus, or clicks the **What's This** button  and then clicks the control. If there is no custom Help topic for the form, the Microsoft Access Help file is displayed.
- If you enter the context ID of the Help file topic as a positive number, the Help topic will be displayed in a full Help topic window. If you add a minus sign (–) in front of the context ID, the Help topic will be displayed in a pop-up window. It is important to note that the context ID does not have a negative number when authored in Microsoft Help Workshop. You must add the minus sign when setting the property to make the topic display in the pop-up window.

[Add the **What's This** button to a form](#)

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to display the form's property sheet.
3. In the **WhatsThisButton** property box, click **Yes**.

Note If you add the **What's This** button to a form, but didn't specify a custom Help file in the **HelpFile** property box for the form, the Microsoft Access Help file is displayed when the user clicks the **What's This** button  and then clicks the form or a control on the form.

[Show All](#)

Create help text for a button or command

Some of the content in this topic may not be applicable to some languages.

 [Create a custom ToolTip \(ScreenTip\) for a toolbar button](#)

[ToolTips](#) appear when the mouse pointer rests on a button or [combo box](#) on a toolbar.

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If the toolbar isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar name you want to show.
3. With the **Customize** dialog box open, right-click the toolbar button you want to create a tip for, and then click **Properties** on the shortcut menu.
4. In the **ScreenTip** property box, type the tip that you want to display.

Notes

- If you don't specify a custom ToolTip, Microsoft Access by default displays the text specified in the **Caption** property.
- If the ToolTip doesn't appear when the mouse pointer rests over the button or combo box, verify that you have selected the **Show ScreenTips on Toolbars** box on the **Options** tab of the **Customize** dialog box.

 [Create "What's This" tips for toolbar buttons or menu commands](#)

"What's This" tips appear when you click **What's This?** on the **Help** menu or press SHIFT+F1, and then click a toolbar button or menu command.

 [Create the tip](#)

1. Create the source file using the text editor or word-processing program of your choice. Make sure to save the source file in Rich Text Format. For information on creating the source file, see the Help that comes with the

Windows Help Compiler.

2. Compile the Help file using the Windows Help Compiler.

Note For best results, put the Help file in the same folder as your application. That way, each site using your application can use the same setting for the **HelpFile** property, but install the application in the folder of their choice. If you omit a path setting in this property, Microsoft Access looks for the Help file in the folder where your application is installed.

[Assign the tip to a toolbar button or menu command](#)

1. On the **View** menu, point to **Toolbars**, and then click **Customize**.
2. If the toolbar isn't already shown within the program window, click the **Toolbars** tab, and then double-click the toolbar name you want to show.
3. With the **Customize** dialog box open, right-click the toolbar button or menu command you want to assign a tip to, and then click **Properties** on the shortcut menu.
4. In the **HelpFile** property box, specify the path and file name for the compiled Help file.
5. In the **HelpContextID** property box, type the number of the topic (other than 0).

Notes

- You must map the values set for the **HelpContextID** property to the context strings in your Help system. For more information, see the Help file that comes with the Microsoft Windows Help Compiler.
- The Windows Help Compiler is available on the Microsoft Software Library (MSL).
- You can't display custom "What's This" tips through the Office Assistant.

[Show All](#)

I can't activate Microsoft Office

[You might not have an Internet or e-mail account set up on your computer](#)

To activate or renew your subscription to Microsoft electronically, you must have a modem and an Internet or e-mail account. You can get an account through an Internet service provider or through your corporate network. If you don't have an Internet or e-mail connection, then you can register by telephone by following the instructions in the Office Activation Wizard.

[The network might be busy](#)

Check your network connection and try to activate or renew you subscription later, when there's less network activity. You can also register by telephone by following the instructions in the Office Activation Wizard.

[Show All](#)

Set the default working folder

You can change the default working folder for all your Microsoft Office programs by renaming the My Documents folder, moving the folder to a new location, or both.

Note In versions prior to Microsoft Windows 2000, the My Documents folder was located in the [root directory](#). In Windows XP and Windows 2000, My Documents is a desktop folder and is represented by an icon of a folder with a sheet of paper in it. To move your My Documents folder off of the desktop, look in Windows XP or Windows 2000 Help for additional information.



[Change the default working folder for Office](#)

1. Save and close all files, and then quit any programs that you have running.
2. Do one of the following:
 - To rename the My Documents folder, right-click the My Documents folder in My Computer, and then click **Rename** on the shortcut menu. Type the new name for the folder, and then press ENTER.
 - To move the My Documents folder, drag the My Documents folder to a new location in the My Computer folder.



[Change the default working folder for Microsoft Access only](#)

1. On the **Tools** menu, click **Options**, and then click the **General** tab.
2. In the **Default database folder** box, type the path for the folder you want to display as the default working folder.

For example, type **c:\work**.

[Show All](#)

Create a network place

Before you can create a [Web folder](#), you should see your system administrator for the [URL](#) of a Web server that you can save files to.



[From your Desktop](#)

1. In **My Network Places**, double-click **Add Network Place**.
2. Follow the instructions in the Add Network Place Wizard.



[From within Microsoft Windows Explorer](#)

- In Windows Explorer, click **My Network Places** under **Folders**, double-click **Add Network Place** and follow the instructions in the Add Network Place Wizard.



[From within Microsoft Access](#)

1. Do one of the following:
 - On the **File** menu, click **New** to display the [New File task pane](#).
 - In the **Open** dialog box (**File** menu), select **My Network Places** under **Look in**.
 - In the **Save As** dialog box (**File** menu), type in the name of the [data access page](#), select **My Network Places** under **Save in**.
2. Double-click **Add Network Place** , select **Create a new Network Place**, click **Next**, and then follow the instructions in the Add Network Place Wizard.

[Show All](#)

About sending e-mail messages

You can set message options to determine how and when the output of a [database object](#) or [data access page](#) is delivered, and the response required from the recipient, in the following ways:

Set the importance and sensitivity level You can set the level of importance for a message to indicate if the recipient needs to read your message immediately. If you want a reply to your message by a certain date and time, you can set a flag to remind the recipient to respond. You can also mark the sensitivity of the message content as private, personal, or confidential.

Set the security level If you and the recipients of your messages use Microsoft Exchange Server, you can sign a message with a [digital signature](#), or seal (encode) a message to help ensure that the message is not readable to anyone other than the recipients that you select.

Track messages and replies You can track your messages by receiving notification when your message is delivered and when it's read by the recipient. To easily find, sort, filter, or group messages, you can assign a keyword or phrase to a message. You can also add voting buttons to a message so that others vote on an issue through e-mail.

Set delivery options You can use an expiration date to show that the content of a message is no longer up-to-date. You can have the message saved in a specified folder in Microsoft Outlook after you send it. You can also specify if you want another recipient to receive replies to your message.

About tracking work and finding files with Microsoft Outlook

Microsoft Outlook is a messaging and collaboration client you can use to send mail; schedule meetings, events, and appointments; store a list of contacts; maintain a task list for yourself and assign tasks to others; and track your activities. If you've installed Outlook, you can use some of its features from within your other Microsoft Office programs.

If Outlook isn't available, click **Programs** on the **Start** menu, and then click **Microsoft Outlook** to install it.

- With the Outlook Journal, you can automatically track all the files you create in any Office program. For example, if you need to track databases based on when they're created and revised instead of where they're located, you can have Outlook create a journal entry each time you work on a database.
- If you want to track only one file in an Office program or track files in programs other than Office, you can manually record each activity you perform on the file or files in the Outlook Journal.
- You can link tasks, journal entries, spreadsheets, documents, databases, and presentations to a person or business in your contact list. For example, if you want to track contacts responsible for working on a particular database, you can link contact information to the database and other related items, such as spreadsheets, e-mail messages, and appointments.
- If you have installed the Microsoft Outlook Integrated File Management system, you can use Outlook to find files for any Office program, files on the network, and Outlook items. You can sort, group, and change the view for your search results, and these results can be used to easily access relevant files. Search results can also be saved as a file you can share with others.

[Show All](#)

Track work and find files with Outlook

By using the Journal feature in Microsoft Outlook, you can automatically track all files you create in any Microsoft Office program. For example, if you need to track databases based on when they're created and revised instead of where they're located, you can have Outlook create a journal entry each time you work on a database.

[Automatically track databases in the Outlook Journal](#)

1. Start Outlook.
2. On the **Tools** menu, click **Options**.
3. Click the **Preferences** tab, and then click **Journal Options**.
4. In the **Also record files from** list, select the **Microsoft Access** check box.

Notes

- Databases that are automatically recorded in the Outlook Journal sometimes take a few minutes to appear in the Journal.
- If Microsoft Access does not appear in the **Also record files from** dialog box on a multi-user computer, the logon profile for one or more users does not contain registry information for the Office programs installed on the computer. For information on how to set up the computer so that the Office programs appear in the **Also record files from** box for a particular user, search the applicable Knowledge Base articles on the [Microsoft Developer Network](#) Web site.

[Manually record work on any file in the Outlook Journal](#)

1. Make sure the file you want to record work for is not open.
2. Start Outlook.
3. On the Outlook Bar, click **Other** or **Other Shortcuts**.
4. Double-click **My Computer**  and then locate the file you want to

record.

If **My Computer** is not available, you need to install the Microsoft Outlook Integrated File Management system.

5. On the Outlook Bar, click **My Shortcuts** and make sure that **Journal**  is visible.
6. Drag the file you want to record onto **Journal** .
7. Select the options you want, and then click **Save and Close** .

[Use Microsoft Outlook to find files or Outlook items](#)

- Click the Windows **Start** button, point to **Search** (or **Find**), and then click **Using Microsoft Outlook**.

Note To search for all files on your computer, you must have the Microsoft Outlook Integrated File Management system installed on your computer.

[Show All](#)

Copy a file

1. Click **Open** under the **File** menu.
2. In the **Look In** box, click the drive or folder that contains the file you want to copy.
3. In the folder list, double-click folders until you open the folder that contains the file you want.

If you can't find the file in the folder list, you can search for it.

[How?](#)

1. Click **Tools**, and then click **Search**.
2. Do one of the following:

[Search for a file containing specified text](#)

You can find a file containing text in its title, contents, or properties.

1. On the **Basic** tab, in the **Search text** box type the text to search for in Office files, Web pages, and Outlook items.

[Tip](#)

You can use wildcards in the **Search text** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. To limit where to search, in the **Search in** box, select one or more drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single location, type it directly in the **Search in** box.
3. To limit the types of search results, in the **Results should be** box, select the types of items to find. To find all types of files, Web pages, and Outlook items, select **Anything**.

4. Click **Search**.



To view all properties of a found item, click the action button following the item, and then choose **Properties**.



1. On the **Advanced** tab, enter one or more search criteria.



1. In the **Property** box, choose a property from the list or type in a property name. The **Property** box displays the properties available in the currently open document.



You can use wildcards in the **Property** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. In the **Condition** box, choose a condition from the list, or type in a condition.
3. In the **Value** box, enter the value to associate with the condition.
4. If previous search criteria exist, click **and** to add a criterion that must be true in addition to previous criteria. Click **or** to add a criterion that is sufficient regardless of previous criteria.
5. Click **Add** to add the search criterion.

Note If the value is invalid for the condition or property, the **Add** button is unavailable.

2. To limit where to search, in the **Search in** box, select one or more

drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single drive, folder, or Web site, type it directly in the **Search in** box.

3. To limit the types of search results, in the **Results should be** box, select the types of files, Web pages, and Outlook items to find. To find all types of files and items, select **Anything**.
4. Click **Search**.



[Tip](#)

To view all properties of a found item, click the action button following the item, and then choose **Properties**.

4. Right-click the file you want to copy, and then click **Copy** on the shortcut menu.
5. In the **Look In** box, click the drive or folder you want to copy the file to.
6. In the folder list, double-click folders until you open the folder you want to copy the file to.
7. Right-click in the folder list (make sure a file is not selected), and then click **Paste** on the shortcut menu.

The **Paste** command is not available if a file is selected in the folder list.

[Show All](#)

Delete a file

1. Click **Open** under the **File** menu.
2. In the **Look In** box, click the drive or folder that contains the file you want to delete.
3. In the folder list, double-click folders until you open the folder that contains the file you want.

If you can't find the file in the folder list, you can search for it.

[How?](#)

1. Click **Tools**, and then click **Search**.
2. Do one of the following:

[Search for a file containing specified text](#)

You can find a file containing text in its title, contents, or properties.

1. On the **Basic** tab, in the **Search text** box, type the text to search for in Office files, Web pages, and Outlook items.

[Tip](#)

You can use wildcards in the **Search text** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. To limit where to search, in the **Search in** box select one or more drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single location, type it directly in the **Search in** box.
3. To limit the types of search results, in the **Results should be** box, select the types of items to find. To find all types of files, Web pages, and Outlook items, select **Anything**.

4. Click **Search**.



To view all properties of a found item, click the action button following the item, and then choose **Properties**.

 [Search for a file or Outlook item based on one or more properties](#)

1. On the **Advanced** tab, enter one or more search criteria.



1. In the **Property** box, choose a property from the list or type in a property name. The **Property** box displays the properties available in the currently open document.



You can use wildcards in the **Property** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. In the **Condition** box, choose a condition from the list, or type in a condition.
3. In the **Value** box, enter the value to associate with the condition.
4. If previous search criteria exist, click **and** to add a criterion that must be true in addition to previous criteria. Click **or** to add a criterion that is sufficient regardless of previous criteria.
5. Click **Add** to add the search criterion.

Note If the value is invalid for the condition or property, the **Add** button is unavailable.

2. To limit where to search, in the **Search in** box, select one or more

drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single drive, folder, or Web site, type it directly in the **Search in** box.

3. To limit the types of search results, in the **Results should be** box, select the types of files, Web pages, and Outlook items to find. To find all types of files and items, select **Anything**.
4. Click **Search**.



[Tip](#)

To view all properties of a found item, click the action button following the item, and then choose **Properties**.

4. Right-click the file you want to delete, and then click **Delete** on the shortcut menu.

[Show All](#)

Move a file

1. Click **Open** under the **File** menu.
2. In the **Look In** box, click the drive or folder that contains the file you want to move.
3. In the folder list, double-click folders until you open the folder that contains the file you want.

If you can't find the file in the folder list, you can search for it.

[How?](#)

1. Click **Tools**, and then click **Search**.
2. Do one of the following:

[Search for a file containing specified text](#)

You can find a file containing text in its title, contents, or properties.

1. On the **Basic** tab, in the **Search text** box, type the text to search for in Office files, Web pages, and Outlook items.

[Tip](#)

You can use wildcards in the **Search text** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. To limit where to search, in the **Search in** box, select one or more drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single location, type it directly in the **Search in** box.
3. To limit the types of search results, in the **Results should be** box, select the types of items to find. To find all types of files, Web pages, and Outlook items, select **Anything**.

4. Click **Search**.



To view all properties of a found item, click the action button following the item, and then choose **Properties**.



1. On the **Advanced** tab, enter one or more search criteria.



1. In the **Property** box, choose a property from the list or type in a property name. The **Property** box displays the properties available in the currently open document.



You can use wildcards in the **Property** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. In the **Condition** box, choose a condition from the list, or type in a condition.
3. In the **Value** box, enter the value to associate with the condition.
4. If previous search criteria exist, click **and** to add a criterion that must be true in addition to previous criteria. Click **or** to add a criterion that is sufficient regardless of previous criteria.
5. Click **Add** to add the search criterion.

Note If the value is invalid for the condition or property, the **Add** button is unavailable.

2. To limit where to search, in the **Search in** box, select one or more

drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single drive, folder, or Web site, type it directly in the **Search in** box.

3. To limit the types of search results, in the **Results should be** box, select the types of files, Web pages, and Outlook items to find. To find all types of files and items, select **Anything**.
4. Click **Search**.



[Tip](#)

To view all properties of a found item, click the action button following the item, and then choose **Properties**.

4. Right-click the file you want to move, and then click **Cut** on the shortcut menu.
5. In the **Look In** box, click the drive or folder you want to move the file to.
6. In the folder list, double-click folders until you open the folder you want to move the file to.
7. Right-click in the folder list (make sure a file is not selected), and then click **Paste** on the shortcut menu.

The **Paste** command is not available if a file is selected in the folder list.

[Show All](#)

Rename a file

1. Click **Open** under the **File** menu.
2. In the **Look In** box, click the drive or folder that contains the file you want to rename.
3. In the folder list, double-click folders until you open the folder that contains the file you want.

If you can't find the file in the folder list, you can search for it.

[How?](#)

1. Click **Tools**, and then click **Search**.
2. Do one of the following:

[Search for a file containing specified text](#)

You can find a file containing text in its title, contents, or properties.

1. On the **Basic** tab, in the **Search text** box, type the text to search for in Office files, Web pages, and Outlook items.

[Tip](#)

You can use wildcards in the **Search text** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. To limit where to search, in the **Search in** box select one or more drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single location, type it directly in the **Search in** box.
3. To limit the types of search results, in the **Results should be** box, select the types of items to find. To find all types of files, Web pages, and Outlook items, select **Anything**.

4. Click **Search**.



To view all properties of a found item, click the action button following the item, and then choose **Properties**.

 [Search for a file or Outlook item based on one or more properties](#)

1. On the **Advanced** tab, enter one or more search criteria.



1. In the **Property** box, choose a property from the list or type in a property name. The **Property** box displays the properties available in the currently open document.



You can use wildcards in the **Property** box. Type a question mark (?) to match any single character, or type an asterisk (*) to match any number of characters. For example, s?t finds "sat" and "set"; s*d finds "sad" and "started."

2. In the **Condition** box, choose a condition from the list, or type in a condition.
3. In the **Value** box, enter the value to associate with the condition.
4. If previous search criteria exist, click **and** to add a criterion that must be true in addition to previous criteria. Click **or** to add a criterion that is sufficient regardless of previous criteria.
5. Click **Add** to add the search criterion.

Note If the value is invalid for the condition or property, the **Add** button is unavailable.

2. To limit where to search, in the **Search in** box, select one or more

drives, folders, Web sites, or Outlook mailboxes. To search everywhere, select **Everywhere**. To specify a single drive, folder, or Web site, type it directly in the **Search in** box.

3. To limit the types of search results, in the **Results should be** box, select the types of files, Web pages, and Outlook items to find. To find all types of files and items, select **Anything**.
4. Click **Search**.



[Tip](#)

To view all properties of a found item, click the action button following the item, and then choose **Properties**.

4. Right-click the file you want to rename, and then click **Rename** on the shortcut menu.
5. Type the new name, and then press ENTER.

[Show All](#)

Remove personal or hidden information

Before you give others a copy of your database, project, or data access page, it's a good idea to review personal and hidden information, and decide whether it's appropriate to include. For example, if you entered personal information, such as your name or e-mail address, when you registered your software, some Microsoft Office documents store that information as part of the file. Information contained in file properties and any custom properties that you add to the file, such as an "author" or "owner" field, are also not automatically removed.

You may want to remove personal information from the file and from the file properties before you share it with others.

[Remove personal information from an Access file](#)

1. On the **Tools** menu, click **Options**.
2. On the **General** tab, select the **Remove personal information from this file** check box.

[Remove personal information from a data access page](#)

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. If the property sheet isn't visible, click **Properties**  on the **Page Design** toolbar.
4. On the **Other** tab, set the **RemoveFileProperties** property to **True**.

Your documents and Microsoft

Unless you send your document to Microsoft, Microsoft does not have access to any information in your document.

[Show All](#)

Print a record, datasheet, or database object

[Print the datasheet of a table, query, or form](#)

1. Open the table, query, or form in [Datasheet view](#), [PivotTable view](#), or [PivotChart view](#).
2. Do one of the following:

1. **Print the entire datasheet, or print the data in PivotTable view or PivotChart view**

1. If your datasheet contains a [subdatasheet](#) that you also want to print, click the [expand indicator](#) (+) to the left of each row whose subdatasheet you want to expand.

Caution If you expand all subdatasheets (click **Format**, point to **Subdatasheet**, and then click **Expand All**), Access might return an unexpectedly large number of records. Don't print after expanding all subdatasheets unless you are certain that you want to print all of the resulting records.

2. Do one of the following:
 - To change the settings in the **Print** dialog box before printing, click **Print** on the **File** menu, and then select the options you want.
 - To print immediately without changing the settings in the **Print** dialog box, click **Print**  on the toolbar.

1. Print only certain records in the datasheet

1. Select the records you want to print.

[How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .
Multiple records	The record selector of the first record, and then drag to extend the selection.
All records	Select All Records on the Edit menu.

2. On the **File** menu, click **Print**, and then click **Selected Record(s)**.

[Print a report](#)

1. Do one of the following:

Select the report in the [Database window](#).

Open the report in [Design view](#), [Print Preview](#), or Layout Preview.

2. On the **File** menu, click **Print**.
3. Enter the settings you want in the **Print** dialog box.

Under **Printer**, specify a printer.

Under **Print Range**, specify all pages or a range of pages.

Under **Copies**, specify the number of copies and whether they should be collated.

[Print a form](#)

[Print an open form](#)

1. On the **File** menu, click **Print**.
2. In the **Print** dialog box, select the printing options you want.

The output depends on the view in which the form is open, except in Design view. If the form is open in Design view, it prints in Form view.

[Print a form from the Database window](#)

1. Click **Forms**  under **Objects**.
2. Select the form you want to print.
3. On the **File** menu, click **Print**.
4. In the **Print** dialog box, select the printing options you want.

Microsoft Access prints the form in the view specified in the form's **DefaultView** property.

[Print selected records in Form view or Datasheet view](#)

1. Open the form in [Form view](#) or [Datasheet view](#).
2. Select the records by clicking the [record selector](#). To select multiple records, click the first [record selector](#) and drag through to the last record you want to print.

Note If the record selector is not visible, press SHIFT+SPACEBAR. If you're in Edit mode, press F2 before pressing SHIFT+SPACEBAR.

3. On the **File** menu, click **Print**.
4. In the **Print** dialog box under **Print Range**, click **Selected Record(s)**.

[Print a data access page](#)

1. Open a [data access page](#) in [Page view](#) or in Microsoft Internet Explorer.
2. Locate the record or records that you want to print. If necessary, click the [expand indicator](#) (+) to expand or collapse any groups you do or do not want to print.
3. On the **File** menu, click **Print**.

Note The data access page might contain a [PivotTable list](#) or [Spreadsheet Component](#), which has additional printing considerations. For more information, click **Help**  on the toolbar for the PivotTable list or spreadsheet.

[Print a database object from a shortcut](#)

After you create a shortcut for a table, query, form, or report, you can print the object directly from the shortcut icon.

- Right-click the shortcut, and then click **Print** on the [shortcut menu](#).

[Tip](#)

You can quickly print from a shortcut by dragging the shortcut icon to a printer icon on the desktop or in the Printers folder.

[Show All](#)

Print mailing labels

1. If the labels do not exist, create them.

[How?](#)

Before creating labels that print on a dot-matrix or tractor-feed printer, set the default printer and paper size.

1. In the [Database window](#), click **Reports** .
2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click **Label Wizard**.
4. Click the table or [query](#) that contains the data for the labels, and then click **OK**.
5. Follow the instructions in the wizard.

If the resulting label report doesn't look the way you want, you can customize it in [Design view](#).

2. Click **Print** on the toolbar.

If the printed labels don't look the way you want, customize the page settings in the **Page Setup** dialog box.

[Show All](#)

Print the design of a database or a database object (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In a [Microsoft Access database](#), you can use the Print Relationships Wizard to create a [report](#) that displays the [relationships](#) as they appear in the [Relationships window](#). You can also use the Documenter to print the design characteristics, such as the properties or the permissions, of a [database object](#).

[Print the Relationships window](#)

1. Open the Access database whose relationships you want to print.
2. On the **Tools** menu, click **Relationships** to open the Relationships window.
3. On the **File** menu, click **Print Relationships**.

Note In a [Microsoft Access project](#), the **Print Relationships** menu command is not available. However, you can print database diagrams.

[Tip](#)

The Print Relationships Wizard generates a report with the default name and header **Relationships for *databasename***; the header contains the date that you created the report. You can save this report for future reference.

[Print the design characteristics of a database object](#)

1. On the **Tools** menu, point to **Analyze**, and then click **Documenter**.
2. Do one of the following:

Click the tab that corresponds to the type of database object you want to view or print.

Click the **All Object Types** tab to display a complete list of objects in the

database.

3. Select the objects whose definition you want to view or print.
4. Click **Options** to specify which features of the selected object you want to print, and then click **OK**.
5. To generate the report that displays the definitions, click **OK**.
6. Do one of the following:

To print the definitions, click **Print**  on the toolbar. You might want to check the length of your definitions before printing them, because some definitions, particularly those for forms and reports, can be many pages long.

To output the definitions to an [HTML](#) file, a Microsoft Excel worksheet, a Rich Text Format file, an MS-DOS text file, or a file in [Report Snapshot](#) format, click **Export** on the **File** menu, and then complete the **Export Report 'ReportName' To** dialog box.

[Show All](#)

Print part of a string as group header

1. Open a grouped report in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar, and make sure that the properties in the **Sorting And Grouping** box are set as follows:

Property	Setting
GroupOn	Prefix Characters
GroupInterval	(Type the number of characters)

3. Click the **Text Box**  tool on the [toolbox](#) and click the [group header](#).
4. Select the control and then click **Properties**  on the toolbar.
5. In the **ControlSource** property box, type the [expression](#) (by using the **Left** or **Right** function) that will extract the appropriate characters. In a report in a [Microsoft Access database](#) (.mdb), you can click the **Build** button  to use the [Expression Builder](#) to create the expression.

 [Example of extracting the first alphabet of a string](#)

If you're tracking office equipment and furniture, and the asset number includes an alphabetic code for distinguishing one type of asset from another (for example, FR100 for furniture), you could use an expression similar to the ones in the following table.

This expression	Returns
=Left([AssetCode], 2)	FR
=Right([AssetCode], 3)	100

Note The Alphabetical List of Products report in the Northwind sample database has an example of printing the first letter of each group of products in the group header. To view this report, open the Northwind database in the Samples subfolder of your Office folder, and then open the Alphabetical List of Products report in Design view.

[Show All](#)

Print and preview a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Printing a database diagram gives you a useful picture of your database structure to refer to or distribute. You can preview the page breaks in the database diagram. You can also further customize the way your printed database diagram will look by changing the layout of the database diagram, and by specifying the table views in the database diagram.

[Print a database diagram](#)

1. Display the diagram and make sure its layout is what you want to see on the printed page.
2. From the **File** menu, choose **Print**.
3. Depending on your printer setup, you may need to make or verify a selection in the **Print** dialog box and click **OK**.

The Database Diagram Designer displays a set of grid lines showing how the database diagram will be broken into pages when it prints. You can recalculate the grid lines after you've added or rearranged objects.

[Preview page breaks](#)

- If grid lines aren't already displayed, choose **View Page Breaks** from the **Diagram** menu.

[Recalculate page breaks](#)

- Right-click in the diagram, and choose **Recalculate Page Breaks** from the shortcut menu.

[Show All](#)

Set page setup options for printing

[For a table, query, form, or report](#)

1. Open the [database object](#) in one of the following views:

Open a table or [query](#) in [Datasheet view](#), [Print Preview](#), [PivotTable view](#), or [PivotChart view](#).

Open a form or report in any view.

2. On the **File** menu, click **Page Setup**.
3. Click the following tabs and set the options you want:

Tab	Options
Margins	Whether to print data only (for forms and reports), or to print headings (for tables or queries)
Page	Orientation, paper size, and printer
Columns	Number, size, and layout of columns (for forms and reports)

[For a data access page](#)

1. Open a data access page in [Design view](#) or [Page view](#).
2. On the **File** menu, click **Page Setup**.
3. In the **Page Setup** dialog box, set options for paper size, headers and footers, page orientation, and margins.

[Set options for a module](#)

1. Open the module in [Design view](#).
2. On the **File** menu, click **Print**.
3. In the **Print** dialog box, click **Setup**.
4. Set the options you want.

Notes

- Page setup options aren't available for macros.

- Microsoft Access stores the settings for page setup options with a form or report, so you set these options only once for each form or report. For tables, queries, data access pages, and modules, you must set page setup options every time you print.

[Show All](#)

Set default print margins

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab.
3. Under **Print margins**, specify the margins you want to use as defaults.

Notes

- Changing these options doesn't affect the margin settings for existing forms and reports.
- These settings do not apply to [data access pages](#).

[Show All](#)

Don't print page header and footer on first and last pages of a report

1. Open the report in [Design view](#).
2. Double-click the [report selector](#) to open the property sheet.
3. Set the **PageHeader** or **PageFooter** property to one of the following settings:

All Pages (Default) Prints the [page header](#) or [footer](#) on all pages.

Not With Rpt Hdr Doesn't print the page header or footer on the page that has the [report header](#).

Not With Rpt Ftr Doesn't print the page header or footer on the page that has the [report footer](#).

Not With Rpt Hdr/Ftr Doesn't print the page header or footer on a page that has either a report header or footer.

Note When you set the **PageHeader** property to **Not With Rpt Ftr** or **Not With Rpt Hdr/Ftr**, Microsoft Access prints the report footer on a new page.

[Show All](#)

Cancel printing if a report doesn't contain any records

1. Open the report in [Design view](#).
2. Double-click the [report selector](#) to display the report's property sheet.
3. Set the **OnNoData** property to **[Event Procedure]** or to the name of a macro that cancels previewing or printing the report.
 - If you use an event procedure, use the **MsgBox** statement to display a message, and set the **Cancel** argument of the Report_NoData sub procedure to **True**.
 - If you use a macro, use the **MsgBox** action to display a message that describes why there is no data for the report, and use the **CancelEvent** action to cancel printing or previewing the report.

Note The Employee Sales by Country report in the Northwind sample database uses an event procedure in the **OnNoData** property. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Employee Sales by Country report in Design view.

[Show All](#)

Print the group header on the same line as the first detail record

A report that uses block layout prints the [group header](#) on the same line as the first detail record. You can create a report that uses a block layout in one of the following ways:

- For a new report, use the Report Wizard and choose **Block** as the layout.
- For an existing report, hide duplicate data.

[How?](#)

[Hide duplicate data on a report with only sorted records](#)

1. Open a sorted report that might contain duplicate values in [Design view](#).
2. Select the text box that might contain duplicate data and click **Properties**  on the toolbar.
3. Set the **HideDuplicates** property to **Yes**.

In a text box for which **HideDuplicates** is set to **Yes**, Microsoft Access prints a value in the first record in which that value occurs and, if records with the duplicate value continue onto another page, at the top of a new page.

[Hide duplicate data on a report with grouped records](#)

1. Open a grouped report that might contain duplicate values in [Design view](#).
2. Move the text box for the field you're grouping on from the [group header](#) to the [detail section](#).
3. Select the text box and click **Properties**  on the toolbar.
4. Set the **HideDuplicates** property to **Yes**.
5. Remove lines and all other [controls](#) from the group header.
6. Double-click the [section selector](#) of the group header to display its

[property sheet](#).

7. Set the **Height** property of the group header to **0**.

The group name will appear only at the beginning of a new group or at the top of a new page if the group continues for more than one page.

[Show All](#)

Repeat a group header on another page

1. Open the report in [Design view](#).
2. Double-click the [section selector](#) of the [group header](#) to display the property sheet.
3. Set the **RepeatSection** property to **Yes**.

[Show All](#)

Set the option to print each record, group, or section on a separate page

[For a form or report](#)

1. Open the form or report in [Design view](#).
2. Do one or more of the following:

To print each record on a separate page, double-click the [section selector](#) of the [detail section](#), and set the **ForceNewPage** property to **After Section**.

To print each group on a separate page, double-click the section selector of the [group header](#), and set the **ForceNewPage** property to **Before Section**. Alternatively, you can set the **ForceNewPage** property of the [group footer](#) to **After Section**.

To print a section on a separate page, double-click the section selector of a section, and set its **ForceNewPage** property to either **Before Section** or **Before & After**.

Note The **ForceNewPage** property is available for all sections except [page headers](#) and [footers](#).

[For a data access page](#)

1. Open a data access page in [Design view](#).
2. Double-click the [section bar](#) of the section that you want to print.
3. Do one or more of the following:
 - To print each detail record on a separate page, set the **PageBreakBefore** or **PageBreakAfter** property of the header section of the lowest [group level](#) to **Always**.
 - To start each group on a separate page, set the **PageBreakBefore** property of the caption section of the corresponding group level to **Always**. If the group doesn't have a caption section, set the **PageBreakBefore** property of the header section.

- To start a specific section— such as a [record navigation section](#) or a [group footer](#)— on a separate page, set its **PageBreakBefore** property to **Always**.

Note Setting the **PageBreakBefore** and **PageBreakAfter** properties of a section to **Always** ensures that no other section will be printed on the same page as this section.

[Show All](#)

About Access databases

A database is a collection of information that's related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. If your database isn't stored on a computer, or only parts of it are, you may be tracking information from a variety of sources that you're having to coordinate and organize yourself.

For example, suppose the phone numbers of your suppliers are stored in various locations: in a card file containing supplier phone numbers, in product information files in a file cabinet, and in a spreadsheet containing order information. If a supplier's phone number changes, you might have to update that information in all three places. In a database, however, you only have to update that information in one place—the supplier's phone number is automatically updated wherever you use it in the database.

 [Access database files](#)

Using Microsoft Access, you can manage all your information from a single database file. Within the file, you can use:

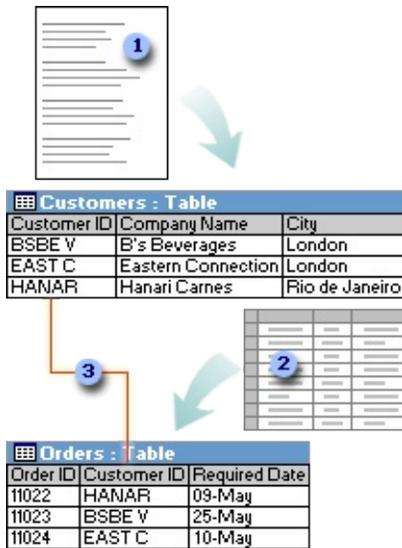
- Tables to store your data.
- Queries to find and retrieve just the data you want.
- Forms to view, add, and update data in tables.
- Reports to analyze or print data in a specific layout.
- Data access pages to view, update, or analyze the database's data from the Internet or an intranet.



- 1 Store data once in one table, but view it from multiple locations. When you update the data, it's automatically updated everywhere it appears.
- 2 Display data in a query
- 3 Display data in a form
- 4 Display data in a report
- 5 Display data in a data access page

[Tables and relationships](#)

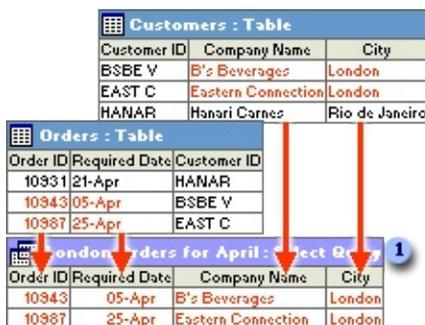
To store your data, create one table for each type of information that you track. To bring the data from multiple tables together in a query, form, report, or data access page, define relationships between the tables.



- 1 Customer information that once existed in a mailing list now resides in the Customers table.
- 2 Order information that once existed in a spreadsheet now resides in the Orders table.
- 3 A unique ID, such as a Customer ID, distinguishes one record from another within a table. By adding one table's unique ID field to another table and defining a relationship, Microsoft Access can match related records from both tables so that you can bring them together in a form, report, or query.

Queries

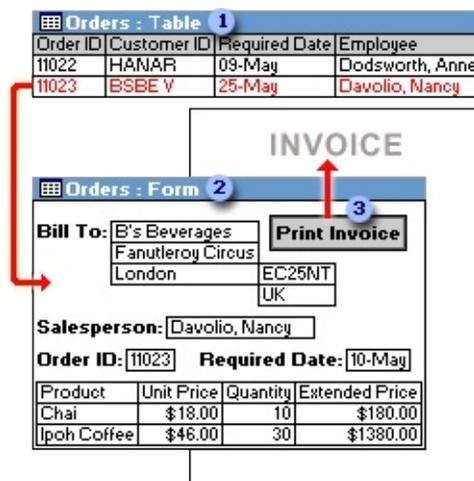
To find and retrieve just the data that meets conditions that you specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform predefined or custom calculations on your data.



1 This query accesses separate tables to retrieve the Order ID, Required Date, Company Name, and City information for customers in London whose orders were required in April.

Forms

To easily view, enter, and change data directly in a table, create a form. When you open a form, Microsoft Access retrieves the data from one or more tables, and displays it on the screen with the layout you choose in the Form Wizard, or with the layout that you created on your own in [Design view](#).



1 A table displays many records at the same time, but you might have to scroll to see all of the data in a single record. Also, when viewing a table, you can't update data from more than one table at the same time.

2 A form focuses on one record at a time, and it can display fields from more than one table. It can also display pictures and other objects.

3 A form can contain a button that prints, opens other objects, or otherwise automates tasks.

Reports

To analyze your data or present it a certain way in print, create a report. For example, you might print one report that groups data and calculates totals, and another report with different data formatted for printing mailing labels.



- 1 Use a report to create mailing labels.
- 2 Use a report to show totals in a chart.
- 3 Use a report to calculate totals.

[Data access pages](#)

To make data available on the Internet or an intranet for interactive reporting, data entry, or data analysis, use a data access page. Microsoft Access retrieves the data from one or more tables and displays it on the screen with the layout that you created on your own in Design view, or with the layout you chose in the Page Wizard.



- 1 Click the expand indicator ...
- 2 ... to display the data and record navigation toolbar for the next level of detail.

3 Use the record navigation toolbars to move to, sort, and filter records, and to get Help.

[Show All](#)

About designing a database

Good database design ensures that your database is easy to maintain. You store data in tables and each table contains data about only one subject, such as customers. Therefore, you update a particular piece of data, such as an address, in just one place and that change automatically appears throughout the database.

A well-designed database usually contains different types of queries that show the information you need. A query might show a subset of data, such as all customers in London, or combinations of data from different tables, such as order information combined with customer information.

Customers : Table		
Customer ID	Company Name	City
BSBE V	B's Beverages	London
EAST C	Eastern Connection	London
HANAR	Hanari Carnes	Rio de Janeiro

Orders : Table		
Order ID	Required Date	Customer ID
10931	21-Apr	HANAR
10943	05-Apr	BSBE V
10987	25-Apr	EAST C

London Orders for April : Select Query				
Order ID	Required Date	Company Name	City	
10943	05-Apr	B's Beverages	London	
10987	25-Apr	Eastern Connection	London	

1 This query retrieves the order ID, company name, city, and required date information for customers in London whose orders were required in April.

The results you want from your database— - the forms and [data access pages](#) you want to use, and the reports you want to print— - don't necessarily provide clues about how you should structure the tables in your database, because you often base forms, reports, and data access pages on queries instead of tables.

Before you use Microsoft Access to actually build tables, queries, forms, and other objects, it's a good idea to sketch out and rework your design on paper first. You can also examine well-designed databases similar to the one you are designing, or you can open the Northwind sample database and then open the [Relationships window](#) to examine its design.

Follow these basic steps when designing your database.

[Determine the purpose of your database](#)

The first step in designing a database is to determine its purpose and how it's to be used:

- Talk to people who will use the database. Brainstorm about the questions you and they would like the database to answer.
- Sketch out the reports you'd like the database to produce.
- Gather the forms you currently use to record your data.

As you determine the purpose of your database, a list of information you want from the database will begin to emerge. From that, you can determine what facts you need to store in the database and what subject each fact belongs to. These facts correspond to the fields (columns) in your database, and the subjects that those facts belong to correspond to the tables.

[Determine the fields you need in the database](#)

Each field is a fact about a particular subject. For example, you might need to store the following facts about your customers: company name, address, city, state, and phone number. You need to create a separate field for each of these facts. When determining which fields you need, keep these design principles in mind:

- Include all of the information you will need.
- Store information in the smallest logical parts. For example, employee names are often split into two fields, FirstName and LastName, so that it's easy to sort data by LastName.
- Don't create fields for data that consists of lists of multiple items. For example, in a Suppliers table, if you create a Products field that contains a comma-separated list of each product you receive from the supplier, it will be more difficult to find only the suppliers that provide a particular product.
- Don't include derived or calculated data (data that is the result of an [expression](#)). For example, if you have a UnitPrice field and a Quantity field, don't create an additional field that multiplies the values in these two fields.

- Don't create fields that are similar to each other. For example, in a Suppliers table, if you create the fields Product1, Product2, and Product3, it will be more difficult to find all suppliers who provide a particular product. Also, you will have to change the design of your database if a supplier provides more than three products. You need only one field for products if you put that field in the Products table instead of in the Suppliers table.

[Determine the tables you need in the database](#)

Each table should contain information about one subject. Your list of fields will provide clues to the tables you need. For example, if you have a HireDate field, its subject is an employee, so it belongs in the Employees table. You might have a table for Customers, a table for Products, and a table for Orders.

[Determine which table each field belongs to](#)

When you decide which table each field belongs to, keep these design principles in mind:

- Add the field to only one table.
- Don't add the field to a table if it will result in the same information appearing in multiple records in that table. If you determine that a field in a table will contain a lot of duplicate information, that field is probably in the wrong table.

For example, if you put the field containing the address of a customer in the Orders table, that information will probably be repeated in more than one record, because the customer will probably place more than one order. However, if you put the address field in the Customers table, it will appear only once. In this respect, a table in a [Microsoft Access database](#) differs from a table in a flat file database such as a spreadsheet.

When each piece of information is stored only once, you update it in one place. This is more efficient, and it also eliminates the possibility of duplicate entries that contain different information.

[Identify the field or fields with unique values in each record](#)

In order for Microsoft Access to connect information stored in separate tables — for example, to connect a customer with all the customer's orders— each table in your database must include a field or set of fields that uniquely identifies each individual record in the table. Such a field or set of fields is called a [primary key](#).

[Determine the relationships between tables](#)

Now that you've divided your information into tables and identified [primary key](#) fields, you need a way to tell Microsoft Access how to bring related information back together again in meaningful ways. To do this, you define [relationships](#) between tables.

You may find it useful to view the relationships in an existing well-designed database such as the Northwind sample database.

[Refine your design](#)

After you have designed the tables, fields, and [relationships](#) you need, it's time to study the design and detect any flaws that might remain. It is easier to change your database design now than it will be after you have filled the tables with data.

Use Microsoft Access to create your tables, specify relationships between the tables, and enter enough sample data in your tables so you can test your design. To test the relationships in your database, see if you can create queries to get the answers you want. Create rough drafts of your forms and reports and see if they show the data you expect. Look for unnecessary duplications of data and eliminate them.

[Enter data and create other database objects](#)

When you are satisfied that the table structures meet the design principles described here, then it's time to go ahead and add all your existing data to the tables. You can then create other database objects— - [queries](#), [forms](#), [reports](#), [data access pages](#), [macros](#), and [modules](#).

[Use Microsoft Access analysis tools](#)

Microsoft Access includes two tools that can help you refine the design of a [Microsoft Access database](#).

- The Table Analyzer can analyze the design of one table at a time, can propose new table structures and [relationships](#) if appropriate, and can divide a table into new related tables if that makes sense.
- The Performance Analyzer can analyze your entire database and make recommendations and suggestions for improving it. The wizard can also implement these recommendations and suggestions.

[Show All](#)

Create an Access database

Microsoft Access provides three methods to create an [Access database](#).

[Create a database by using a Database Wizard](#)

You can use a Database Wizard to create in one operation the required tables, forms, and reports for the type of database you choose— this is the easiest way to start creating your database. The wizard offers limited options to customize the database.

1. Click **New**  on the toolbar.
2. In the **New File** task pane, under **Templates**, click **On my computer**.
3. On the **Databases** tab, click the icon for the kind of database you want to create, and then click **OK**.
4. In the **File New Database** dialog box, specify a name and location for the database, and then click **Create**.
5. Follow the instructions in the Database Wizard.

Note You can't use the Database Wizard to add new tables, forms, or reports to an existing database.

[Create a database by using a template](#)

This is the quickest way to create a database. This method works best if you can find and use a template that very closely matches your requirements.

1. Click **New**  on the toolbar.
2. In the **New File** task pane, under **Templates**, either search for a specific template or click **Templates home page** to browse for a suitable template.
3. Click on the template you want, and click **Download**.

[Create a database without using a Database Wizard](#)

You can create a blank database and then add the tables, forms, reports, and other objects later— this is the most flexible method, but it requires you to

define each database element separately. Either way, you can modify and extend your database at any time after it has been created.

1. Click **New**  on the toolbar.
2. In the **New File** task pane, under **New**, click **Blank Database**.
3. In the **File New Database** dialog box, specify a name and location for the database, and then click **Create**.

The [Database window](#) appears, and you can create the [objects](#) that you want in your database.

[Show All](#)

Improve performance of an Access database

Depending on your computer's configuration and your working environment, there are several things you can do that may improve the performance of Microsoft Access or your [Microsoft Access database](#).

If you are optimizing the performance of an Access database, the best place to start is with the Performance Analyzer. You can use the Performance Analyzer to analyze a whole database or just selected objects in a database. The Performance Analyzer can also make some proposed changes for you if you want.

 [Run the Performance Analyzer](#)

You can use the Performance Analyzer to optimize the performance of a [Microsoft Access database](#). The Performance Analyzer is not available in an [Access project](#).

1. Open the Access database you want to optimize.
2. On the **Tools** menu, point to **Analyze**, and then click **Performance**.
3. Click the tab for the type of [database object](#) that you want to optimize. Click the **All Object Types** tab to view a list of all database objects at once.
4. Select the names of the database objects that you want to optimize. Click **Select All** to select all database objects in the list.
5. Repeat steps 3 and 4 until you've selected all the objects you want to optimize, and then click **OK**.

The Performance Analyzer lists three kinds of analysis results: **Recommendation**, **Suggestion**, and **Idea**. When you click an item in the **Analysis Results** list, information about the proposed optimization is displayed in the **Analysis Notes** box below the list. **Suggestion** optimizations have potential tradeoffs that you should consider before performing them. To view a description of the trade-offs, click a **Suggestion** in the list and then read the information in the **Analysis Notes** box. Access can perform **Recommendation** and **Suggestion** optimizations

for you. You must perform **Idea** optimizations yourself.

6. Click one or more of the **Recommendation** or **Suggestion** optimizations you want performed, and then click **Optimize**. The Performance Analyzer will perform the optimizations and then mark them as **Fixed**. Continue this process until the Performance Analyzer has completed all **Recommendations** and **Suggestions** that you want it to perform.

To perform all **Recommendation** and **Suggestion** optimizations, click **Select All**, and then click **Optimize**.

To perform an **Idea** optimization, click the optimization, and then follow the instructions displayed in the **Analysis Notes** box.

Note The Performance Analyzer doesn't provide suggestions on how to improve the performance of Microsoft Access itself or the system it is running on.

 [Improve the performance of Microsoft Access and your system](#)

The following guidelines can help you optimize the performance of Microsoft Access on your computer:

- If you're using databases that other users don't need to share, install Microsoft Access and all your databases on your hard disk drive rather than on a network server.
- If you are the only person using a database, open the database for exclusive use; in the **Open** dialog box, click the arrow next to **Open** button, and then click **Open Exclusive**.
- Make more memory available by closing applications that you aren't using.
- Increase RAM on your computer. 40 MB of memory is recommended— 32 MB of memory plus an additional 8 MB of memory for Microsoft Access.
- Don't use any of your RAM for a RAM disk.
- Periodically delete unnecessary files and empty your Recycle Bin, compact your databases, and then defragment your hard disk with the Microsoft Windows Disk Defragmenter. To run the Disk Defragmenter, click the Windows **Start** button, point to **Programs**, point to **Accessories**, point to **System Tools**, and then click **Disk Defragmenter**.
- In most cases, the default virtual memory setting used by Windows should perform optimally. However, in some situations, adjusting virtual memory

parameters can improve performance. If you've already tried deleting unnecessary files and you still have a performance problem, try changing the default virtual memory setting in the following cases:

- You don't have much disk space available on the drive that is currently being used for virtual memory, and another local drive with space is available.
- Another local drive is available that is faster than the current drive (unless that disk is heavily used).

In these cases, you might get better performance by specifying a different drive for virtual memory.

You also might get better performance by specifying that the disk space available for virtual memory be at least 25 MB minus available RAM. For example, if your computer has 12 MB of RAM, you should specify at least 13 MB of virtual memory. You may want to specify more if you are running several large applications.

To change Windows virtual memory parameters, in Windows Control Panel, double-click the System icon, click **Performance Options** under the **Advanced** tab, click the **Change** button in the Virtual Memory section, and then specify a different hard disk, or enter a value in the **Initial size** box that is at least 25 minus your available RAM.

- If you have a wallpaper (full-screen background) bitmap on your Windows desktop, replace it with a solid color or pattern bitmap, or no bitmap at all.
- If you use a screen saver, use a blank screen saver or consider not using one at all.
- Eliminate disk compression software, or consider moving your databases to an uncompressed drive.
- To ensure optimal performance, use substitution fonts only if you need dual-font support to display all of your data.

[Improve table performance](#)

- Use the Performance Analyzer to analyze specific tables in your database.
- Design tables without redundant data. A well-designed database is a prerequisite for fast data retrieval and updates. If existing tables contain

redundant data, you can use the Table Analyzer Wizard to split your tables into related tables to store your data more efficiently.

- Choose appropriate [data types](#) for fields; this can save space in your database and improve [join](#) operations. When defining a field, choose the smallest data type or field size that's appropriate for the data in the field.
- Create [indexes](#) for fields you sort, join, or set criteria for. You can make dramatic improvements in the speed of queries by indexing fields on both sides of joins, or by creating a [relationship](#) between those fields and indexing any field used to set criteria for the query. Finding records through the **Find** dialog box is also much faster when searching an indexed field.

Indexes aren't appropriate in all cases, however. Indexes add to the size of the .mdb file, reduce concurrency (the ability of more than one user to modify a page at the same time) in multiuser applications, and decrease performance when you update data in fields that are indexed or when you add or delete records. It's a good idea to experiment to determine which fields should be indexed. Adding an index may speed up a query by one second, but slow down adding a row of data by two seconds and cause locking problems. Or it may add negligible gains depending on which other fields are indexed. For example, adding an index to a PostalCode field may provide very little performance gain if a CompanyName field and LastName field in the table are already indexed. Regardless of the types of queries you create, you should only index fields that have mostly unique values.

- In a multiple-field index, use only as many fields in the index as necessary.

[Improve performance of linked tables](#)

Note You can link a table only in a [Microsoft Access database](#), not a [Microsoft Access project](#).

Although you can use linked tables as if they were regular tables in your Microsoft Access database, it's important to keep in mind that they aren't actually in your Access database. Each time you view data in a linked table, Microsoft Access has to retrieve records from another file. This can take time, especially if the linked table is on a network or in an [SQL database](#).

If you're using a linked table on a network or in an SQL database, follow these guidelines for best results:

- You can greatly enhance performance when opening the main database and opening tables and forms by forcing the linked database to remain open. To do this, create an empty table in the linked database, and link the table in the main database. Then use the **OpenRecordset** method to open the linked table. This prevents the Microsoft Jet database engine from repeatedly opening and closing the linked database and creating and deleting the associated .ldb file.
- View only the data that you need. Don't move up and down the page unnecessarily in the datasheet. Avoid jumping to the last record in a large table. If you want to add new records to a large table, use the **Data Entry** command on the **Records** menu to avoid loading existing records into memory.
- Use filters or queries to limit the number of records that you view in a form or datasheet. This way, Microsoft Access can transfer less data over the network.
- In queries that involve linked tables, avoid using functions in query criteria. In particular, avoid using [domain aggregate functions](#), such as **Dsum**, anywhere in your queries. When you use a domain aggregate function, Microsoft Access retrieves all of the data in the linked table to execute the query.
- If you often add records to a linked table, create a form for adding records that has the **DataEntry** property set to **Yes**. When you open the form to enter new data, Microsoft Access doesn't display any existing records. This property setting saves time because Microsoft Access doesn't have to retrieve the records in the linked table.
- Remember that other users might be trying to use an external table at the same time you are. When a Microsoft Access database is on a network, avoid locking records longer than necessary.

[Improve the performance of tables linked to SQL server](#)

If the data in your [front-end/back-end application](#) consists only of Microsoft SQL Server tables, you can use either a Microsoft Access project or a Microsoft Access database as the front end of your application. However, by using an Access project, you will benefit from the advantages of projects such as a larger number of tables and larger database size. If you want the data in your front-

end/back-end application to consist of both Microsoft Access tables and Microsoft SQL server tables, you must use an Access database as the front end.

If you're connecting to an external SQL database table, you can achieve the best performance results by linking to the SQL tables instead of opening the tables directly. You can open external SQL tables directly only by using Microsoft Visual Basic code. Linked tables are considerably faster, more powerful, and more efficient than directly opened tables.

Additional performance tips

- Retrieve only the data you need. Design your queries to limit the number of records that you retrieve, and select only the fields you need, so that Microsoft Access can transfer as little data as possible over the network.
- Don't use updatable result sets (dynasets) if you're not retrieving many records, are not updating the data, and don't need to see changes made by other users. To prevent result sets from being updated in a form, set the form's **RecordsetType** property to **Snapshot**. Snapshots are faster to open and scroll through than dynasets.
- If you need to retrieve a large number of records, a dynaset is faster and more efficient than a snapshot. For example, moving to the end of a snapshot requires the entire result set to be downloaded to the local computer, but with a dynaset only the last screen of data is downloaded to the local computer. In addition, the fastest way to add new records to a table, form, or query is to click **Data Entry** on the **Records** menu. (**Data Entry** isn't available if the **RecordsetType** property is set to **Snapshot**.)
- Use cache memory. If you will reuse the most recent data from the server while the application is running, it's faster to retrieve a single large chunk of data (many rows) and store it in a cache than it is to retrieve many individual rows. Microsoft Access forms and datasheets automatically use a cache. If you are retrieving data by using a **Recordset** object created in Microsoft Visual Basic, you can use the **CacheSize** property to specify how many records to retrieve at one time into local memory.
- For bulk update queries against ODBC data sources, optimize performance on the server by setting the **FailOnError** property to **Yes**.
- Avoid using queries that cause processing to be done on a local client computer. When accessing external data, the Jet database engine processes data locally only when the operation can't be performed by the external database server. Query operations performed locally (as defined by the SQL

commands used to implement them) include:

- JOIN operations between tables from different remote data sources. (For example, assume that the join involves a local table or query with few records and a remote table with many more records, and the remote table's join field is indexed. In this case, Access returns only the records that match the local table or query, thus greatly improving query performance.)
- JOIN operations based on a query with the DISTINCT predicate or a GROUP BY clause.
- [Outer joins](#) containing syntax not supported by the server.
- DISTINCT predicates containing operations that can't be processed remotely.
- The LIKE operator used with Text or Memo fields (may not be supported by some servers).
- Multiple-level GROUP BY arguments and totals, such as those used in reports with multiple grouping levels.
- GROUP BY arguments based on a query with a DISTINCT predicate or a GROUP BY clause.
- Crosstab queries that have more than one aggregate, that have field, row, or column headings that contain aggregates, or that have a user-defined ORDER BY clause.
- TOP *n* or TOP *n* PERCENT predicates.
- User-defined functions, or operators or functions that aren't supported by the server.
- Complex combinations of [INNER JOIN](#), [LEFT JOIN](#), or [RIGHT JOIN](#) operations in nested queries.

[Improve performance in a multiuser environment](#)

The following guidelines can help you optimize the performance of [Microsoft Access databases](#) that are used in a multiuser environment.

- Put only the tables on a network server and keep other database objects on users' computers. The database's performance will be faster because only data is sent across the network. You can separate the tables from other database objects by using the Database Splitter Wizard.
- Choose a suitable record-locking strategy. If you use optimistic locking (**No Locks**), Access doesn't lock a record or page until a user saves changes to it, and data might be more readily available as a result. If you use

pessimistic locking (**Edited Records**), Access locks the record as soon as a user begins to edit it— - records might be locked for longer periods of time, but users don't have to decide whose changes to save if two users edit the same record simultaneously.

- Choose the appropriate locking level strategy (page-level or record-level locking).
- Avoid locking conflicts by adjusting **Refresh Interval**, **Update Retry Interval**, **Number Of Update Retries**, and **ODBC Refresh Interval** settings (if applicable).
- Optimize the performance of tables (including tables linked to other data sources such as Microsoft SQL Server).
- Convert the [workgroup information file](#) from Access 97 or earlier to the current version of Access.

[Improve query performance](#)

[Databases, tables, and indexes](#)

- Compact your database. Compacting can speed up queries because it reorganizes a table's records so that they reside in adjacent database pages ordered by the table's [primary key](#). This will improve the performance of sequential scans of a table's records because the minimum number of database pages will have to be read to retrieve all of the records. After compacting the database, run each query to compile it using the updated table statistics.
- [Index](#) any field used to set [criteria](#) for the query and index fields on both sides of a [join](#), or create a [relationship](#) between these fields. When you create relationships, the [Microsoft Jet database engine](#) creates an index on the [foreign key](#) if one does not already exist; otherwise, it uses the existing index.

The Jet database engine automatically optimizes a query that joins a Microsoft Access table on your hard drive and an [ODBC](#) server table if the Access table is small and the joined fields are indexed. In this case, Access improves performance by requesting only the necessary records from the server. Make sure tables you join from different sources are indexed on the join fields.

- Index the fields you use for sorting.

- Use the **Between...And**, the **In**, and the = operators on indexed fields.

Fields, calculations, and criteria

- When defining a field in a table, choose the smallest [data type](#) appropriate for the data in the field. Also, give fields you'll use in [joins](#) the same or compatible data types, such as AutoNumber and Number (if the **FieldSize** property is set to **Long Integer**).
- When creating a query, add only the fields you need. In fields used to set [criteria](#), clear the **Show** check box if you don't want to display those fields.
- If you use criteria to restrict the values in a field used in a join between tables with a [one-to-many relationship](#), test whether the query runs faster with the criteria placed on the "one" side or the "many" side of the join. In some queries, you get faster performance by adding the criteria to the field on the "one" side of the join instead of the "many" side.
- Avoid [calculated fields](#) in [subqueries](#). If you add a query that contains a calculated field to another query, the [expression](#) in the calculated field may slow performance in the top-level query. In the example below, query Q1 is used as the input for query Q2:

```
Q1: SELECT IIF([MyColumn]="Yes","Order Confirmed","Order Not Cor  
Q2: SELECT * FROM Q1 WHERE X="Order Confirmed";
```

Because the **Iif** expression in Q1 cannot be optimized, Q2 also cannot be optimized. If an expression that can't be optimized is nested within a subquery, the entire query cannot be optimized.

An alternative way to construct the query is as follows:

```
Q1: SELECT * FROM MyTable WHERE MyColumn = "Yes";
```

- If expressions are necessary in the output, try to place them in a [control](#) on a form or report. For example, you could change the previous query into a parameter query that prompts for the value of MyColumn, and then base a form or report on the query. On the form or report, you could then add a [calculated control](#) that displays "Order Confirmed" or "Order Not Confirmed" depending on the value in MyColumn.

Construct the query as follows:

```
PARAMETERS [To see confirmed orders, enter Yes. To see
unconfirmed orders, enter No.] Text;
SELECT *
FROM MyTable
WHERE MyColumn = [To see confirmed orders, enter Yes. To see
unconfirmed orders, enter No.];
```

In the calculated control on the form or report, enter:

```
=IIF([MyColumn]="Yes","Order Confirmed","Order Not Confirmed")
```

- When grouping records by the values in a joined field, specify **Group By** for the field that's in the same table as the field you're totaling (calculating an [aggregate](#) on). For example, if you create a query that totals the Quantity field in the Order Details table and groups by OrderID, it's recommended that you specify **Group By** for the OrderID field in the Order Details table. If you specify **Group By** for the OrderID field in the Orders table, Microsoft Access must join all the records first and then perform the aggregate, instead of performing the aggregate and then joining only the necessary fields.

For greater speed, use **Group By** on as few fields as possible. As an alternative, use the **First** function where appropriate.

If a totals query includes a join, consider grouping the records in one query and adding this query to a separate query that will perform the join. This improves performance in some queries.

- Avoid restrictive query criteria on calculated and [nonindexed](#) fields whenever possible.

[Expressions and Rushmore optimization](#)

[Overview of Rushmore optimization](#)

Microsoft Access can optimize simple [expressions](#) or complex expressions in the **Criteria** row of the query [design grid](#), or in a WHERE clause in an SQL SELECT statement. For certain types of complex expressions, Access can use Rushmore, a data access technology used by the [Microsoft Jet database engine](#), to achieve a greater level of optimization. In addition, the **Count** function is highly optimized for queries using Rushmore.

Rushmore queries will work with Access tables, as well as with Microsoft FoxPro and dBASE tables (.dbf files). You can't use Rushmore with [ODBC data sources](#), since Access sends these queries to the ODBC data source instead of processing them locally.

[Simple optimizable expressions](#)

A simple optimizable [expression](#) can form an entire expression or can appear as part of an expression. A simple optimizable expression takes one of the following forms:

IndexedField ComparisonOperator Expression

[Order Date] = #09/15/96#

or

Expression ComparisonOperator IndexedField

#11/1/96# < [HireDate]

In a simple optimizable expression:

- *IndexedField* can either be a field that has its own [index](#) or a field that's the first field in a multiple-field index.
- *ComparisonOperator* must be one of the following: <, >, =, <=, >=, <>, **Between...And, Like, In.**
- *Expression* can be any valid expression, including constants, functions, and fields from other tables.

If you have created indexes for the LastName, Age, and HireDate fields in the Employees table, the following are simple optimizable expressions:

[LastName] = "Smith"

[Age] >= 21

#12/30/90# < [HireDate]

Employees.[LastName] = Customers.[LastName]

[LastName] In ("Smith", "Johnson", "Jones")

[Age] Between 18 And 65

[Complex expressions](#)

A complex [expression](#) is created by combining two simple expressions with the **And** or the **Or** operator. A complex expression takes one of the following forms:

*SimpleExpression **And** SimpleExpression*

or

*SimpleExpression **Or** SimpleExpression*

A complex expression is fully or partially optimizable depending on whether one or both simple expressions are optimizable, and which operator you used to combine them. A complex expression is Rushmore-optimizable if all three of the following are true:

- The expression uses **And** or **Or** to join two [criteria](#).
- Both criteria are made up of simple optimizable expressions.
- Both expressions contain [indexed](#) fields. The fields can be indexed individually or they can be part of a multiple-field index.

Note You can optimize multiple-field indexes if you query the indexed fields in the order they appear in the [Indexes window](#), beginning with the first indexed field and continuing with adjacent fields (up to and including 10 fields). For example, if you have a multiple-field index that includes the LastName and FirstName fields, you can optimize a query on LastName or on LastName and FirstName, but you can't optimize a query on FirstName.

The following different combinations of simple expressions in query criteria indicate the level of query optimization.

Fully optimizable query result (using Rushmore)

- (optimizable expression) **And** (optimizable expression)
- (optimizable expression) **Or** (optimizable expression)

Partially optimizable query result (doesn't use Rushmore)

- (optimizable expression) **And** (not optimizable expression)

Not optimizable query result

- (not optimizable expression) **And** (not optimizable expression)
- (not optimizable expression) **Or** (not optimizable expression)
- **Not** (optimizable expression)
- **Not** (not optimizable expression)

You can also use parentheses to group combinations of simple expressions. The preceding rules also apply to combinations of expressions grouped with parentheses.

Once you've combined simple optimizable expressions into complex expressions, these complex expressions can, in turn, be combined to form even more complex expressions that might be optimizable according to the preceding rules.

These examples assume that you created indexes for the LastName and HireDate fields (simple expressions using them are optimizable) but not the MiddleInitial or FirstName fields (simple expressions using them are not optimizable). The following examples of combined simple expressions show the extent to which the result is optimized.

Fully optimizable query result (using Rushmore)

```
[LastName] = "Smith" And [Hire Date] < #12/30/90#
```

```
[LastName] = "Smith" Or [Hire Date] < #12/30/90#
```

Partially optimizable query result (doesn't use Rushmore)

[LastName] = "Smith" And [MiddleInitial] = "C"

[FirstName] = "Terry" And [MiddleInitial] = "C"

Not optimizable query result

[LastName] = "Smith" Or [MiddleInitial] = "C"

[FirstName] = "Terry" Or [MiddleInitial]= "C"

[General suggestions](#)

- If the **RecordSource** property for a form or report is set to an [SQL statement](#), save the SQL statement as a query, and then set the **RecordSource** property to the name of the query.
- For bulk update queries against [ODBC data sources](#), optimize performance on the server by setting the **FailOnError Property** to **Yes**.
- If your data doesn't change often, use [make-table queries](#) to create tables from your query results. Use the resulting tables rather than queries as the basis for your forms, reports, or other queries, and make sure you add [indexes](#) according to the guidelines recommended here.
- Avoid using [domain aggregate functions](#), such as the **DLookup** function, to access data from a table that's not in the query. Domain aggregate functions are specific to Microsoft Access, which means that the [Microsoft Jet database engine](#) can't optimize queries that use them. Instead, add to the query the table that the function was accessing, or create a [subquery](#).
- If you are creating a [crosstab query](#), use fixed column headings whenever possible.

[Improve filter performance](#)

If the lists in fields in the [Filter By Form](#) window take too long to display or they aren't displaying values from the underlying table, you can change Filter By Form performance for all tables, queries, and forms, or change performance for a specific form. In either case, you can prevent the lists from displaying the underlying table's field values, display field values on the list for certain types of [indexed](#) or nonindexed fields only, or change the record limit that determines if

the list displays a field's values.

[Improve Filter By Form performance for all tables, queries, and forms in the database](#)

These settings affect all tables and queries, and all text box [controls](#) whose **FilterLookup** property is set to **Database Default**.

1. On the **Tools** menu, click **Options**.
2. Click the **Edit/Find** tab, and then set the **Filter By Form Defaults** according to the performance you want to achieve.
 - If the list of values takes too long to display in nonindexed fields only, try limiting the lists to indexed fields. You can do this by clearing the **Local Nonindexed Fields** and **ODBC Fields** check boxes. If the lists take too long to display in indexed fields, clear the **Local Indexed Fields** check box as well since there are too many records in the indexes for the lists to display quickly.

Note If you use the same nonindexed field repeatedly to filter records, consider indexing it to improve filtering and other search operations on the field.

- If lists aren't displaying the values from indexed or nonindexed fields, check under **Show List Of Values In** to make sure the appropriate boxes are selected, or try increasing the number in the **Don't Display Lists Where More Than This Number Of Records Read** box so that it's greater than or equal to the maximum number of records in any nonindexed field in the underlying table.

[Improve Filter By Form performance for a specific form](#)

For each text box [control](#) on a form, you can specify whether the values stored in the field appear on its list in the Filter By Form window.

1. Open a form in [Design view](#).
2. Display the property sheet for any text box control.

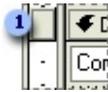
[How?](#)

1. Open a table, query, form, report, or [data access page](#) in [Design view](#).
2. Open the property sheet by doing one of the following:
 - On a form or report, double-click the [form selector](#) or the [report selector](#).



1 Form or report selector

- On a data access page, click **Select Page** on the **Edit** menu.
- On a table or query, click **Properties**  on the toolbar.
- For a section on a form or report, double-click the [section selector](#).



1 Section selector

- For a section on a data access page, double-click the section bar.



1 Section bar

- To set properties of a query field, click the cell in the **Field** row, and then click **Properties**  on the toolbar.
 - To set properties of a query field list, click anywhere in the list, and then click **Properties**  on the toolbar.
 - To set properties for the body of a data access page, click above the first section, and then click **Properties**  on the toolbar.
 - To set properties for a [control](#), click the control, and then click **Properties**  on the toolbar.
3. In the property sheet, click the property you want to set, and then do one of the following:
 - If an arrow appears in the property box, click it and then select a value from the list.

- Type a setting or [expression](#) in the property box.
- If a **Build** button  appears next to the property box, click it to display a [builder](#) or list of builders.

[Tips](#)

- To get Help on any property in the property sheet, click the property and then press F1.
- If you need more space to enter or edit a property setting, press SHIFT+F2 to open the **Zoom** box.

Note If the **AllowDesignChanges** property of a form is set to **All Views**, you can also set form properties in [Form view](#) and [Datasheet view](#).

3. In the **FilterLookup** property box, set the property to the setting that will achieve the performance you want.
 - If the list takes too long to display in the Filter By Form window, either set the **FilterLookup** property to **Never**, or set it to **Database Default** and then make the appropriate changes in the **Options** dialog box (**Tools** menu).
 - If the list in the Filter By Form window isn't displaying the values stored in that field, make sure the **FilterLookup** property isn't set to **Never**. If the **FilterLookup** property is set to **Database Default** or **Always**, make the appropriate changes in the **Options** dialog box (**Tools** menu).

[Improve find and replace performance](#)

An index helps Microsoft Access find and sort records faster. Access uses indexes in a table as you use an index in a book: to find data, it looks up the location of the data in the index. You can create indexes based on a single field or on multiple fields. Multiple-field indexes enable you to distinguish between records in which the first field may have the same value.

Deciding which fields to index

You'll probably want to index fields you search frequently, fields you sort, or fields that you [join](#) to fields in other tables in [queries](#). However, indexes can slow down some [action queries](#) such as [append queries](#), when the indexes for many fields need to be updated while performing these operations.

The [primary key](#) of a table is automatically indexed, and you can't index a field whose [data type](#) is [OLE Object](#). For other fields, you should consider indexing a field if all the following apply:

- The field's data type is Text, Number, Currency, or Date/Time.
- You anticipate searching for values stored in the field.
- You anticipate sorting values in the field.
- You anticipate storing many different values in the field. If many of the values in the field are the same, the index may not significantly speed up queries.

Multiple-field indexes

If you think you'll often search or sort by two or more fields at a time, you can create an index for that combination of fields. For example, if you often set criteria for LastName and FirstName fields in the same query, it makes sense to create a multiple-field index on both fields.

When you sort a table by a multiple-field index, Microsoft Access sorts first by the first field defined for the index. If there are records with duplicate values in the first field, Microsoft Access sorts next by the second field defined for the index, and so on.

You can include up to 10 fields in a multiple-field index.

[Improve form and subform performance](#)

There are several things you can do to make your forms run faster.

[General tips](#)

- Avoid overlapping controls.
- Don't sort records in an underlying [query](#) unless record order is important, especially with multiple-table queries.
- If the underlying [record source](#) includes many records and you want to use the form primarily to enter new records, set the **DataEntry** property of the form to **Yes** so that the form opens to a blank record. If you open a form with all records showing, Microsoft Access has to read each record before it can display the blank record at the end of the recordset.
- Close forms that aren't being used.

[Tips for forms that contain pictures and objects](#)

- Use bitmaps and other graphic objects sparingly.
- Convert unbound object frames that display graphics to image controls.
- Use black-and-white rather than color bitmaps.

[Tips for forms that contain code](#)

A form without a [form module](#) loads more quickly and occupies less disk space. If a form or controls on the form don't use [event procedures](#), the form doesn't require a form module.

- Eliminate code from forms that don't need it by setting the **HasModule** form property to **No**. The **HasModule** property specifies whether the form has a form module.
- Instead of using command buttons with event procedures, you can use command buttons with [macros](#), or [hyperlinks](#).

Notes

- If a form currently contains event procedures, and you decide to eliminate all event procedure code from that form, you must set the **HasModule** property to **No** to completely remove the form module.
- You can still use code with a form that has no form module by calling **Function** procedures from a standard module using an [expression](#). (You can't use **Sub** procedures, because they can't be called using an expression.) To do this, define a **Function** procedure in a standard module and then call that function from an [event property](#) of the form or a control. For example, to use a command button to call a function to open a form, add an expression to the **OnClick** property of the command button like this:
=OpenDataEntry().



[Tips for subforms](#)

- Base subforms on queries rather than on tables, and include only fields from the [record source](#) that are absolutely necessary. Extra fields can decrease subform performance.
- [Index](#) all the fields in the subform that are linked to the [main form](#).
- Index any subform fields that are used for criteria.
- Set the subform's **AllowEdits**, **AllowAdditions**, and **AllowDeletions** properties to **No** if the records in the subform aren't going to be edited. Or set the **RecordsetType** property to **Snapshot**.



[Improve list box and combo box performance](#)

There are several things you can do to make your list boxes, combo boxes, and

[drop-down list boxes](#) run faster:

- Base the list box, combo box, or drop-down list box on a saved [query](#) instead of an SQL statement. On a form, if you use a wizard to create a list box or combo box, Microsoft Access automatically sets the **RowSource** property of the control to an SQL statement. To change the **RowSource** property to a saved query, click the **Build** button next to the **RowSource** property box. With the **Query Builder** window open, click **Save** on the **File** menu, and type a name for the query. When you close the Query Builder window, click **Yes** when Microsoft Access asks if you want to update the property.
- Base the drop-down list box on a saved query instead of a table. On a data access page, if you use a wizard to create a drop-down list box, Access automatically sets the **ListRowSource** property of the control to the table you specified in the wizard. To retrieve rows from a query instead of a table, select the query name from the **ListRowSource** property box.
- In the query specified in the **RowSource** or **ListRowSource** property box, include only fields that are absolutely necessary. Extra fields can decrease performance.
- [Index](#) both the first field displayed in the combo box, list box, or drop-down list box and the bound field (if the fields are different).
- In combo boxes on a form, set the **AutoExpand** property to **No** if you don't need the fill-in-as-you-type feature.
- On a form, if the **AutoExpand** property for a combo box is set to **Yes**, the first displayed field should have a **Text** data type instead of a Number data type. In order to find a match in the list, Microsoft Access converts a numeric value to text. If the data type is **Text**, Microsoft Access doesn't have to do this conversion.
- On a form, if the bound field in a lookup combo box is not the displayed field:
 - Don't use [expressions](#) for the bound field or the displayed field.
 - Don't use restrictions in the row source.
 - Use single-table (or query) row sources rather than multiple-table row sources, if possible.
- Don't create list boxes, combo boxes or drop-down list boxes based on data in a linked table if the data won't change. It's better to import the data into your database in this case.

Note The Query Builder is not available in a data access page. Wizards are not

available in a stand-alone data access page, or when you point a data access page to a database other than the one currently open. In both cases, you will have to create a list box or drop-down list box on your own.

[Improve report and subreport performance](#)

There are several things you can do to make your reports print faster.

[General tips](#)

- Avoid overlapping controls.
- Avoid sorting and grouping on expressions.
- [Index](#) fields you sort or group on.
- Avoid using [domain aggregate functions](#). Include the field in the report's underlying query or use a [subreport](#).
- Make sure the report's underlying query is optimized.
- Use the **HasData** property or NoData event to determine if a report is bound to an empty recordset.

[Tips for reports that contain pictures and objects](#)

- Use bitmaps and other graphic objects sparingly.
- Convert [unbound object frames](#) that display graphics to [image controls](#).
- Use black-and-white rather than color bitmaps.

[Tips for subreports](#)

- Base subreports on queries rather than tables, and include only fields from the record source that are absolutely necessary. Extra fields can decrease subreport performance.
- Index all the fields in the subreport that are linked to the main report.
- Index any subreport fields used for [criteria](#).

[Improve data access page performance](#)

You can do several things to make your data access pages load faster in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

[General tips](#)

- When using a data access page within Microsoft Access, close all database objects that aren't being used. When using a data access page in Internet Explorer, close all windows that aren't being used.
- On data access pages that are used for data entry, set the **DataEntry** property of the page to **True** so that the page opens to a blank record.
- Index any fields that you sort, group, or filter on.
- Bind a data access page to an XML data file, so the page could get its data from a file located on the local client instead of having to access the database server.

[Controls](#)

- Avoid overlapping controls.
- Use bound span controls instead of text boxes to display any data that isn't updateable— for example, to display autonumbered values, values derived from expressions, and hyperlinks.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Pictures and objects](#)

- Use bitmaps and other graphic objects sparingly. Use black-and-white rather than color bitmaps.

[Grouped pages](#)

- Make sure the **ExpandedByDefault** property is set to **False** for the highest group level. Setting it to **False** for lower group levels will speed up other interactions after the page is opened.
- Make sure the **DataPageSize** group level property is set to a low rather than a high number, or is set to **All**. The lower the number, the faster the records

are displayed.

- On a page that contains records with a one-to-many relationship, group records by table rather than by a field or expression.
- On a page that is based on tables with a one-to-many relationship, bind each section to a table rather than to a query. Microsoft Access retrieves the records from a table only as it needs to display them on the page. With a query, Access retrieves all the records before it displays any records on the page. Use a query only when you want to limit the data used on the page, calculate aggregate values, or perform a task that's possible only in a query.

 [Read about making your database an MDE file to improve performance](#)

If your database contains [Microsoft Visual Basic for Applications \(VBA\)](#) code, saving your [Microsoft Access database](#) as an MDE file compiles all modules, removes all editable source code, and compacts the destination database. Your Visual Basic code will continue to run, but it cannot be viewed or edited.

Security Saving your database as an MDE file helps protect your forms and reports without requiring users to log on or requiring you to create and manage the [user accounts](#) and [permissions](#) that are needed for [user-level security](#).

Your database will continue to function normally— you can still update data and run reports. Additionally, the size of the database will be reduced due to the removal of the code, so memory usage is optimized, which will improve performance.

Saving your Access database as an MDE file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in Design view.
- Adding, deleting, or changing references to object libraries or databases.
- Changing code— an MDE file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, queries, [data access pages](#), and macros can be imported from or exported to non-MDE databases.

 [Considerations if you need to modify the design of forms, reports, or modules](#)

Be sure to keep a copy of your original Access database. If you need to modify

the design of forms, reports, or modules in an MDE file, you must modify the original Access database, and then resave it as an MDE file. Saving an Access database containing tables as an MDE file creates complications reconciling different versions of the data if you need to modify the design of the forms, reports, or modules later. For this reason, saving an Access database as an MDE file is most appropriate for the front-end database of a [front-end/back-end application](#).

[Code no longer employs user-level security](#)

In previous versions of Access, user-level security could be used with VBA modules. In Microsoft Access 2000 and later, all Visual Basic code for an Access database (.mdb) file or an [Access project \(.adp\)](#) file, including [stand-alone modules](#) and [class modules](#) (such as code behind forms and reports) must enable the security features by setting a password, or by saving the database as an MDE or ADE file, which removes the source code.

[Consideration before saving your database as an MDE file](#)

Some restrictions may prevent you from saving your Access database as an MDE file:

- You must have password access to the Visual Basic code.
- If your database is replicated, you must first remove [replication](#).
- If your Access database references another Access database, or [add-in](#), you must save all Access databases or add-ins in the chain of references as MDE files.

Additionally, if you define a database password or user-level security before saving an Access database as an MDE file, those features will still apply to an MDE file created from that database. If your Access database has a database password or user-level security defined and you want to remove these features, you must do so before saving it as an MDE file.

To save an Access database that has engaged user-level security as an MDE file, you must meet the following requirements before you can proceed:

- You must join the [workgroup information file](#) that defines the user accounts used to access the database, or that were in use when the database was

created.

- Your user account must have Open/Run and Open Exclusive permissions for the database.
- Your user account must have Modify Design or Administer permissions for any tables in the database, or you must be the [owner](#) of any tables in the database.
- Your user account must have Read Design permissions for all objects in the database.

[About references and MDE files](#)

If you try to create an MDE file from a Microsoft Access database (.mdb) or an [add-in \(.mda\)](#) that references another Access database or add-in, Access displays an error message and doesn't let you complete the operation. To save a database that references another database as an MDE file, you must save all databases in the chain of references as MDE files, starting from the first database referenced. After saving the first database as an MDE file, you must then update the reference in the next database to point to the new MDE file before saving it as an MDE file, and so on.

For example, if Database1.mdb references Database2.mdb, which references Database3.mda, you would proceed as follows:

1. Save Database3.mda as Database3.mde.
2. Open Database 2.mdb and change its reference to point to the new Database3.mde.
3. Save Database2.mdb as Database2.mde.
4. Open Database1.mdb and change its reference to point to the new Database2.mde.
5. Save Database1.mdb as Database1.mde.

[About saving a replicated database as an MDE file](#)

A replicated database (either a [replica](#) or [Design Master](#)) cannot be saved as an MDE file. To save a replicated database as an MDE file, you must first remove [replication](#).

Once a database is saved as an MDE file, it can be replicated; however, replication is only recommended in situations where no further changes need to

be made to the original database. There is no way to make changes to the design of forms, reports, or code in the Design Master MDE file, and design changes made in the original database can't be [synchronized](#) with pre-existing MDE replicas. For this reason, if you need to make a design change to a [replica set](#) of MDE files, you must make that change in the original database, resave it as an MDE file, and then create and distribute an entirely new replica set from the new MDE file.

[Read about compacting an Access database](#)

Microsoft Access combines compacting and repairing an [Access file](#) into a single process.

[Compacting an Access file](#)

If you delete data or objects in an [Access database](#), or if you delete objects in an [Access project](#), the file can become fragmented and use disk space inefficiently. Compacting the Access file makes a copy of the file and rearranges how the file is stored on your disk. Compacting a previous-version Access database won't convert it to Access 2002 - 2003 format.

Compacting optimizes the performance of both Access databases and Access projects. However, in an Access project, compacting does not affect [database objects](#)— such as tables or views— that reside in the [Microsoft SQL Server database](#) instead of in the Access project itself.

Compacting has no effect on autonumbering in an Access project. However, in an Access database, if you have deleted records from the end of a table that has an [AutoNumber field](#), compacting the database resets the AutoNumber value; the AutoNumber value of the next record you add will be one greater than the AutoNumber value of the last undeleted record in the table.

[Repairing an Access file](#)

In most cases, Microsoft Access detects whether an Access file is damaged when you try to open it and gives you the option to repair it at that time. If the current Access file contains a reference to another Access file that is corrupt, Access does not attempt to repair the other file. In some situations, Access may not detect that a file is damaged. If an Access file behaves unpredictably, compact

and repair it.

Access can repair:

- Corruption in a table in an [Access database](#).
- Missing information about the structure of an Access file's Visual Basic for Applications (VBA) project.
- Corruption in a form, report, or module.
- Missing information that Access needs to open a particular form, report, or module.

To help prevent corruption of Access files, follow these guidelines:

- Compact and repair Access files regularly. You can specify that Access compact a file automatically when you close it.
- Back up your Access files regularly.
- Avoid quitting Access unexpectedly. For example, don't quit Access suddenly by turning off your computer.
- If you are experiencing network problems, avoid using a shared Access database located on a network server until the problems are resolved. If possible, move the Access database to a computer where you can access it locally instead of over the network.

[Repairing files after Access shuts down unexpectedly](#)

If a serious problem occurs and causes Microsoft Access to shut down, Access restarts, creates a copy of the file that was open, and names it *filename_Backup.mdb* or *filename_Backup.adp*, where *filename* is the name of the original file. If *filename_Backup.mdb* or *filename_Backup.adp* already exists, Access asks you to specify the name of the file. Access then attempts to compact and repair the original file.

[Show All](#)

Use the Performance Analyzer to optimize an Access database

You can use the Performance Analyzer to optimize the performance of a [Microsoft Access database](#). The Performance Analyzer is not available in an [Access project](#).

1. Open the Access database you want to optimize.
2. On the **Tools** menu, point to **Analyze**, and then click **Performance**.
3. Click the tab for the type of [database object](#) that you want to optimize. Click the **All Object Types** tab to view a list of all database objects at once.
4. Select the names of the database objects that you want to optimize. Click **Select All** to select all database objects in the list.
5. Repeat steps 3 and 4 until you've selected all the objects you want to optimize, and then click **OK**.

The Performance Analyzer lists three kinds of analysis results: **Recommendation**, **Suggestion**, and **Idea**. When you click an item in the **Analysis Results** list, information about the proposed optimization is displayed in the **Analysis Notes** box below the list. **Suggestion** optimizations have potential tradeoffs that you should consider before performing them. To view a description of the tradeoffs, click a **Suggestion** in the list and then read the information in the **Analysis Notes** box. Access can perform **Recommendation** and **Suggestion** optimizations for you. You must perform **Idea** optimizations yourself.

6. Click one or more of the **Recommendation** or **Suggestion** optimizations you want performed, and then click **Optimize**. The Performance Analyzer will perform the optimizations and then mark them as **Fixed**. Continue this process until the Performance Analyzer has completed all **Recommendations** and **Suggestions** that you want it to perform.

To perform all **Recommendation** and **Suggestion** optimizations, click **Select All**, and then click **Optimize**.

To perform an **Idea** optimization, click the optimization, and then follow

the instructions displayed in the **Analysis Notes** box.

Note The Performance Analyzer doesn't provide suggestions on how to improve the performance of Microsoft Access itself or the system it is running on.

[Show All](#)

Open an Access database

1. On the **File** menu, click **Open** .
2. Click a shortcut in the left side of the **Open** dialog box, or in the **Look in** box, click the drive or folder that contains the [Microsoft Access database](#) that you want.
3. In the folder list, double-click folders until you open the folder that contains the database.

If you can't find the database that you want to open, click **Tools** on the toolbar in the **Open** dialog box, and then click **Search** . In the **Search** dialog box, enter additional search criteria.

4. Do one of the following:
 - Double-click the database.
 - To open the database for shared access in a [multiuser](#) environment, so you and other users can read and write to the database, click **Open**.
 - To open the database for read-only access so that you can view it but cannot edit it, click the arrow next to the **Open** button, and then click **Open Read-Only**.
 - To open the database with [exclusive](#) access, click the arrow next to the **Open** button, and then click **Open Exclusive**.
 - To open the database for read-only access and also prevent other users from opening it, click the arrow next to the **Open** button, and then click **Open Exclusive Read-Only**.

Note You can directly open a data file in an external file format such as dBASE, Paradox, Microsoft Exchange, or Microsoft Excel. You can also directly open any [ODBC data source](#), such as Microsoft SQL Server or Microsoft FoxPro. Access automatically creates a new Access database in the same folder as the data file and adds links to each table in the external database.

[Tips](#)

- To open one of the last several databases you had open, click the file name at the bottom of the **File** menu. Microsoft Access opens the database with

the same option settings it had the last time you opened it. If the list of recently used files isn't displayed, click **Options** on the **Tools** menu, click the **General** tab, and then select the **Recently used file list** check box.

- To view a list of shortcuts to databases that you have opened previously, click **History** on the left side of the **Open** dialog box.
- To quickly locate a file or folder in the Favorites folder, click **Favorites** on the left side of the **Open** dialog box. To open a favorite file or folder, click the shortcut and then click **Open**.

Open a sample database or project

1. On the **Help** menu, point to **Sample Databases**, and then click the database or project that you want to open.
2. If the database or project isn't currently installed, click **Yes** to install it now.

Notes

- If you don't see a list of samples when you point to **Sample Databases**, you can install them by double-clicking **Add or Remove Programs** in Windows Control Panel and running Office 2003 Setup.
- If you are logged on to a computer where you do not have administrator permissions, the sample files will be read-only. This is because the user account does not have permissions to edit files. Copy the sample files to a location where you have permissions to edit, and then try opening the files from the new location.

[Show All](#)

Create a shortcut to open an Access file or database object

[Create a shortcut to an Access file in the Favorites folder](#)

The Favorites folder stores shortcuts to folders and files you use frequently, including those in remote locations, so that you don't have to search for the files each time. The original file or folder doesn't move when you create a shortcut to it.

1. Click **Open**  on the toolbar.
2. To create a shortcut to a folder, subfolder, [Microsoft Access database](#), or [Microsoft Access project](#) that appears in the list under the **Look in** box, select the file or folder, click **Tools**, and then click **Add to Favorites**.

[Create a shortcut to open an Access file or database object](#)

You can create a shortcut to open a [database object](#) in a [Microsoft Access file](#) that is stored locally on your computer or remotely on a network file server or in a shared directory.

1. Under **Objects** in the [Database window](#), click the type of database object you want to create a shortcut for.
2. Right-click the object, and then click **Create Shortcut**.
3. If you want the shortcut to go to a location other than the desktop, type a new path in the **Location** box, or click **Browse** to choose the location and have Access fill in the path for you. For example, you might want to store the shortcut in the Favorites folder, which stores shortcuts to items you use frequently.

If you're creating a shortcut for an object in a database that is on a network, Access selects **This Database is on the Network**, and fills in the network path in the **Full Network Path** box. If you move the database later, you can type in a new network path in this format:

\\server\share\filename

With the network path specified, you can, for example, send the shortcut through e-mail so that others with access to the network can use the shortcut, too.

4. Click **OK**.

When you double-click the shortcut, Access opens the database in which the object is stored and displays the object. A shortcut to a [macro](#) or stored procedure runs that macro or stored procedure. To open the object in a specific view, right-click the shortcut, and then click the view you want.

Notes

- If you move the database after you create the shortcut, delete the shortcut and create a new one.
- To delete a shortcut, click it and then press the DELETE key. Deleting the shortcut does not delete the object that the shortcut opens.



You can also drag the object from the Database window to the desktop or to a folder. Make sure the place you want to drag the shortcut to is visible. (If you previously maximized the Access window, minimize it enough to show the desktop or the folder where you want to place the shortcut.)

[Show All](#)

Copy or move an Access file

1. Close the [Microsoft Access file](#). If you are using an [Access database](#) in a [multiuser](#) environment, confirm that all users have closed the file.
2. In My Computer, find the database (.mdb) or [project](#) (.adp) file and then click it.

To select more than one file to copy or move, hold down the CTRL key and click each file.

3. Do one of the following:
 - To copy the file, click **Copy** on the **Edit** menu.
 - To move the file, click **Cut** on the **Edit** menu.
4. Open the folder or disk where you want to put the file or files.
5. On the **Edit** menu, click **Paste**.

To undo the paste operation, click **Undo Copy** or **Undo Move** on the **Edit** menu.

Notes

- When you copy or move an Access project file, you are not copying or moving the [Microsoft SQL Server database](#) to which the Access project is connected.
- If you copy or move an Access database, and another database includes links to objects in that database, use the Linked Table Manager in the other database to update the link information. Otherwise, Access will produce an error when you try to use those linked objects in the other database.



If the destination where you're placing the file is already visible, you can quickly move the file by dragging it to the new destination, or you can copy the file by holding down the CTRL key while you drag the file.

[Show All](#)

Change the default file format

1. Open a [Microsoft Access file](#) if you don't have one open already.
2. On the **Tools** menu, click **Options**, and then click the **Advanced** tab.
3. Under **Default File Format**, click the Access file format you want.

A file in Access 2000 file format can be opened in Access 2000 or later.
You can open a file in Access 2002 - 2003 format only in Access 2002 or later.

[Show All](#)

Delete an Access file

1. Close the [Microsoft Access file](#). If you are using an [Access database](#) in a [multiuser](#) environment, confirm that all users have closed the file.
2. Click **Open**  on the toolbar.
3. In the **Open** dialog box, locate the Access file you want to delete, and then right-click the file.
4. On the [shortcut menu](#), click **Delete**.

Notes

- If you want to restore your file, look in the Recycle Bin. Your deleted file will remain in the Recycle Bin until you empty it.
- When you delete an Access file that contains [data access pages](#), you are not deleting the page HTML (.htm) files.
- When you delete an Access project, you are not deleting the [Microsoft SQL Server database](#) to which the Access project is connected.
- If you delete an Access database, and another database includes links to objects in that database, use the Linked Table Manager in the other database to update the link information. Otherwise, Access will produce an error when you try to use those linked objects in the other database.



To delete an Access file from outside Access, close the file, right-click the file in My Computer, and then click **Delete** on the shortcut menu.

[Show All](#)

Rename an Access file

1. Close the [Microsoft Access file](#). If you are using an [Access database](#) in a [multiuser](#) environment, confirm that all users have closed the file.
2. Click **Open**  on the toolbar.
3. In the **Open** dialog box, locate the Access file you want to rename, and then right-click the file.
4. On the shortcut menu, click **Rename**.
5. Type the new name. Include the .mdb or .adp extension only if you've set up Microsoft Windows to display file extensions for known file types. Press ENTER.

Notes

- A file name can be up to 215 characters, including spaces.
- If you rename a Microsoft Access database, and another database includes links to objects in that database, use the Linked Table Manager in the other database to update the link information. Otherwise, Access will produce an error when you try to use those linked objects in the other database.



You can also rename an Access file from outside Access. Close the file, right-click the file in My Computer, and then click **Rename** on the shortcut menu.

[Show All](#)

About setting a hyperlink base for an Access file

When you create a [hyperlink](#) in a [Microsoft Access file](#), you can make the path to the [destination](#) of the hyperlink a fixed location (absolute link), which identifies the destination by its full address— such as C:\My Documents\Sales.doc— or you can make the path a [relative link](#). You can easily change the path of a relative link by setting the [hyperlink base](#) for the Access file. The value of the **HyperlinkBase** property of an Access file can be a [URL](#) or a location on a network or your hard drive.

For example, you may want hyperlinks to Web pages on an [intranet](#) server in your Access file. If you set the value of the hyperlink base to **http://servername1**, you can create a hyperlink as a relative link, such as **webpage.htm**. Clicking this hyperlink opens the web page located at **http://servername1/webpage.htm**. If you then move the web pages to a different intranet server, you only need to change the value of the **HyperlinkBase** property to **http://servername2** to adjust all of the relative links in the Access file.

[Show All](#)

Set a hyperlink base for an Access file

You can set the path of the [relative links](#) in a [Microsoft Access file](#) by setting the **HyperlinkBase** property.

1. Open the database you want to set a hyperlink base for.
2. On the **File** menu, click **Database Properties**, and then click the **Summary** tab.
3. In the **Hyperlink base** box, type the path of the relative link you want to use for all the hyperlinks you create in this database. The path may point to a location on your hard drive, a network drive, an [intranet](#) or the [Internet](#).

[Show All](#)

View the attributes of an Access file

1. With the [Microsoft Access file](#) open, click **Database Properties** on the **File** menu.
2. Click the **General** tab.

[Show All](#)

Document the summary information of an Access file or page

[Document the summary information of a Microsoft Access file](#)

1. With the [Microsoft Access file](#) open, click **Database Properties** on the **File** menu.
2. Click the **Summary** tab, and then specify as much of the requested information as you want to.
3. Click **OK**.

Note For each item you complete on the **Summary** tab, the associated properties are automatically set.

[Document the summary information of a data access page](#)

1. Open the data access page in [Design view](#) or [Page view](#). If it is a new page that hasn't been saved yet, click **Save** on the **File** menu.
2. On the **File** menu, click **Page Properties**.
3. Set properties as needed on the following tabs.
 - **General** This is the same information that is displayed when you right-click the name of a file in Microsoft Windows Explorer and then click **Properties** on the shortcut menu. The only difference is that when you view these properties in Microsoft Access, the **Attributes** settings are read-only.
 - **Summary** The information that you enter on this tab is added to the SummaryInfo **Document** object in the **Documents** collection. This information is similar to the summary information that you can provide in other Microsoft Office applications. Summary information allows you to better identify a page both from within Microsoft Access and from other programs such as the Microsoft Windows Find Files program. The **Hyperlink Base** setting on the **Summary** tab is used to create the base hyperlink path that is appended to the beginning of relative **HyperlinkAddress** property settings.
 - **Statistics** This information includes the date and time that the active

page was created, the last date and time it was modified, who modified it, the number of times it's been revised, and the total editing time.

- **Contents** This information includes the title and the connection string.
- **Custom** The custom properties that you enter become properties of the **UserDefined Document** object in the **Documents** collection.

[Show All](#)

Create custom properties for use in finding an Access file

1. With the [Microsoft Access file](#) open, click **Database Properties** on the **File** menu.
2. Click the **Custom** tab, and then type values for the new property in the **Name**, **Type**, and **Value** boxes.
3. Click **Add**.
4. When you have finished adding properties, click **OK**.

[Show All](#)

Modify or delete custom properties of an Access file

1. With the [Microsoft Access file](#) open, click **Database Properties** on the **File** menu.
2. Click the **Custom** tab.
3. Click the name of the property you want to delete or modify in the **Name** column in the **Properties** box.
4. To modify the property, enter the changes in the **Type** or **Value** boxes, and then click **Modify**.

To delete the property, click **Delete**.

Note If you try to modify the name of the property, it becomes a new property, and you can only add it. To modify the name of the property, delete the existing property and add it with the new name.

[Show All](#)

Graphic filters and file formats

Microsoft Access can use

You can insert many popular graphics file formats into a [form](#), [report](#), or [data access page](#). This can be done directly or with the use of separate graphics filters by using the **Picture** command on the **Insert** menu.

For forms and reports, you don't need a separate graphics filter installed to insert Enhanced Metafile (.emf), Windows bitmap (.bmp, .rle, .dib), Windows Metafile (.wmf), and icon (.ico) graphics. However, you do need a graphics filter installed to insert all other graphics file formats listed below.

Graphics file formats requiring separate graphics filters in forms and reports

Computer Graphics Metafile (.cgm)	Macintosh PICT (.pct)
CorelDRAW (.cdr)	Portable Network Graphics (.png)
Encapsulated PostScript (.eps)	Tagged Image File Format (.tif)
Graphic Interchange Format (.gif)	WordPerfect Graphics (.wpg)
Hanako (.jsh, .jah, .jbh)	X-Bitmap (.xbm)
JPEG File Interchange Format (.jpg)	

For pages, you don't need a separate graphics filter installed to insert Microsoft Windows bitmap (.bmp, .rle, .dib) graphics. However, you do need a graphics filter installed to insert all other graphic file formats.

The separate graphics filters are not supplied with the stand-alone version of Microsoft Access. To use the filters, you need to install the Microsoft Office Professional Edition, Microsoft Office Premier Edition, or a stand-alone version of Microsoft Word along with your stand-alone version of Access.

Note For more information on each separate graphics filter, see Microsoft Word

Help.

[Show All](#)

About how Access searches for reference libraries

The Visual Basic [project](#) of a [Microsoft Access file](#) contains references to [object libraries](#), and it can also contain references to other files, including other Access files. The locations of referenced files are specified in the **References** dialog box that is available from the **References** command on the **Tools** menu in the [Visual Basic Editor](#). When you open an Access file, if Access doesn't find a referenced file in the specified location, it searches for the reference as follows.

- First, Access searches for a RefLibPaths key in the following location in the Microsoft Windows Registry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Office\11.0\Access
```

If the key exists, Access checks for the existence of a value name that matches the name of the referenced file. If it finds a matching value name, Access loads the reference from the path specified in the corresponding value data.

- If Access doesn't find a RefLibPaths key, it searches for the referenced file in the locations listed below in the following order:
 - Application folder containing the application (the folder where Msaccess.exe is located).
 - Current folder.
 - System folders (the System and System32 folders located in the Windows or WINNT folder).
 - Windows or WINNT folder.
 - PATH environment variable. For more information about environment variables, see Windows Help.
 - The folder that contains the Access file, and any subfolders located in that folder.
- If Access still can't find the reference after performing this search, you must fix the reference manually.

[Show All](#)

Back up and restore an Access database (MDB)

Back up an Access database

1. Save and close all objects in the database.
2. On the **File** menu, click **Back Up Database**.
3. In the **Save Backup As** dialog box, specify the name and location of the backup copy.

Caution If you are using [user-level security](#), you should also create a backup of the [workgroup information file](#). If this file is lost or damaged, you won't be able to start Microsoft Access until you restore or rebuild it.

Restore an Access database from a backup copy

- Depending on what method was used originally to produce the backup copy, use My Computer, the Windows 2000 or later Backup and Recovery Tools, or other backup software to copy the backup Access database to your database folder.

Caution If the existing Access database in the database folder and the backup copy have the same name, restoring the backup copy may replace the existing file. If you want to save the existing file, rename it before you copy the backup database.

Note You can back up individual [database objects](#) by creating a blank database and then importing the objects you want from the original database.

[Show All](#)

About compacting and repairing an Access file

Microsoft Access combines compacting and repairing an [Access file](#) into a single process.

[Compacting an Access file](#)

If you delete data or objects in an [Access database](#), or if you delete objects in an [Access project](#), the file can become fragmented and use disk space inefficiently. Compacting the Access file makes a copy of the file and rearranges how the file is stored on your disk. Compacting a previous-version Access database won't convert it to Access 2002 - 2003 format.

Compacting optimizes the performance of both Access databases and Access projects. However, in an Access project, compacting does not affect [database objects](#)— such as tables or views— that reside in the [Microsoft SQL Server database](#) instead of in the Access project itself.

Compacting has no effect on autonumbering in an Access project. However, in an Access database, if you have deleted records from the end of a table that has an [AutoNumber field](#), compacting the database resets the AutoNumber value; the AutoNumber value of the next record you add will be one greater than the AutoNumber value of the last undeleted record in the table.

Caution When you compact a Microsoft Access file located on a volume that uses the NTFS file system, Access removes the existing file and replaces it with the compacted file. It then applies the default file permissions to the new file. If the file is an [Access database](#), use Access [user-level security](#) instead of file-level permissions. Otherwise, use folder permissions. For more information about the NTFS file system and folder permissions, see Windows Help.

[Repairing an Access file](#)

In most cases, Microsoft Access detects whether an Access file is damaged when

you try to open it and gives you the option to repair it at that time. If the current Access file contains a reference to another Access file that is corrupt, Access does not attempt to repair the other file. In some situations, Access may not detect that a file is damaged. If an Access file behaves unpredictably, compact and repair it.

Access can repair:

- Corruption in a table in an [Access database](#).
- Missing information about the structure of an Access file's Visual Basic for Applications (VBA) project.
- Corruption in a form, report, or module.
- Missing information that Access needs to open a particular form, report, or module.

To help prevent corruption of Access files, follow these guidelines:

- Compact and repair Access files regularly. You can specify that Access compact a file automatically when you close it.
- Back up your Access files regularly.
- Avoid quitting Access unexpectedly. For example, don't quit Access suddenly by turning off your computer.
- If you are experiencing network problems, avoid using a shared Access database located on a network server until the problems are resolved. If possible, move the Access database to a computer where you can access it locally instead of over the network.



[Repairing files after Access shuts down unexpectedly](#)

If a serious problem occurs and causes Microsoft Access to shut down, Access restarts, creates a copy of the file that was open, and names it *filename_Backup.mdb* or *filename_Backup.adp*, where *filename* is the name of the original file. If *filename_Backup.mdb* or *filename_Backup.adp* already exists, Access asks you to specify the name of the file. Access then attempts to

compact and repair the original file.

[Show All](#)

Compact and repair an Access file

To ensure optimal performance, you should compact and repair your Microsoft Access files on a regular basis. Also, if a serious problem occurs while you are working in an Access file and Access attempts to recover it, you might receive a message that the repair operation was cancelled and that you should compact and repair the file.

Caution When you compact a Microsoft Access file located on a volume that uses the NTFS file system, Access removes the existing file and replaces it with the compacted file. It then applies the default file permissions to the new file. If the file is an [Access database](#), use Access [user-level security](#) instead of file-level permissions. Otherwise, use folder permissions. For more information about the NTFS file system and folder permissions, see Windows Help.

You must have Open/Run and Open Exclusive [permissions](#) for an Access database in order to compact and repair it.

[Compact and repair the current Access file](#)

1. If you are compacting a [shared Microsoft Access database](#) that is located on a server or shared folder, make sure that no one else has it open.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Compact and Repair Database/Project**.

[Compact and repair an Access file that is not open](#)

1. Close the current Microsoft Access file. If you are compacting a [shared Access database](#) that is located on a server or shared folder, make sure no one else has it open.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Compact and Repair Database**.
3. In the **Database to Compact From** dialog box, specify the Access file you want to compact, and then click **Compact**.
4. In the **Compact Database Into** dialog box, specify a name, drive, and folder for the compacted Access file.

5. Click **Save**.

If you use the same name, drive, and folder, and the Access database or Access project is compacted successfully, Microsoft Access replaces the original file with the compacted version.

[Compact and repair an Access file automatically every time you close it](#)

Compacting does not occur if you close a [shared Access database](#) while another user has it open.

1. Open the Access database or Access project that you want Microsoft Access to compact automatically.
2. On the **Tools** menu, click **Options**.
3. Click the **General** tab.
4. Select the **Compact on Close** check box.

Note You can stop the compact and repair process by pressing CTRL+BREAK or ESC.

[Show All](#)

Troubleshoot compacting, repairing or recovering an Access file

[I can't compact and repair an Access file.](#)

You can't compact an Access file if any of the following are true.

- You don't have enough storage space on your disk for both the original and compacted versions of the Access database or Access project. Delete any unneeded files and try again.
- You don't have both Open/Run and Open Exclusive [permissions](#) for the Access database. If you don't own this database, contact its owner to find out whether you can be given these permissions. If you own this database, you need to update your permissions.
- Another user has the Access database or Access project open.
- The Access database or Access project is located on a read-only network share, or its file attribute is set to Read Only.

[Compacting an Access file removes the NTFS file permissions.](#)

When you compact a Microsoft Access file located on a volume that uses the NTFS file system, Access removes the existing file and replaces it with the compacted file. It then applies the default file permissions to the new file.

If the file is an [Access database](#), use Access [user-level security](#) instead of file-level permissions. Otherwise, use folder permissions. For more information about the NTFS file system and folder permissions, see Windows Help.

[A log file was created during recovery of an Access file.](#)

After the occurrence of a serious problem such as an unexpected shutdown, if any objects in a Microsoft Access file are deleted or recovered when a Access backs up and recovers the file, Access records this information in a log file with the name *<DestinationFile.log>*, where *DestinationFile* is the name of the new file.

[Show All](#)

About converting an Access file

[When should I convert or enable an Access file?](#)

Whether you should convert or enable a [Microsoft Access file](#) depends on which versions of Access users have upgraded to.

[When to convert from Access 97 or earlier to Access 2002 - 2003 file format](#)

In most cases, you'll want to convert a previous-version [Microsoft Access database](#) to Access 2000 file format if all users have upgraded to Access 2000 or later, and you'll want to convert to Access 2002 - 2003 file format if all users have upgraded to Access 2002 or later. In Access 2002 or later, you can modify data and make design changes to an Access 2000 file. Although you can also [enable](#) an Access 97 or earlier database in Access 2002 or later without converting it, you can change the design of [objects](#) only in the previous version.

Once you've converted an Access database to Microsoft Access 2000 or Access 2002 - 2003 file format, you can't open it in the previous version of Access. You can convert an Access 2000 database back to Access 97. However, you can't convert an Access 2000 database back to Access version 2.0 or 95.

If your Access database is a [multiuser \(shared\) database](#), and all users can't upgrade to Microsoft Access 2000 or later at the same time, you can split the database so that it is a [front-end/back-end application](#). You can then have different versions of the front end connected to the back end, which remains unaltered. Users of Access 2000 or later can use a converted version of the front end, where they can take advantage of new features.

[When to convert an Access 2000 file to Access 2002 - 2003 file format](#)

If you want to save the Access file as an [MDE](#) or [ADE](#) in Access 2002 or later, you must convert the file to Access 2002 - 2003 file format. Otherwise, you don't necessarily need to convert a Microsoft Access 2000 file to the Access 2002 - 2003 file format. If you're converting Access databases from Access 97 or

earlier, it's recommended that you convert to Access 2000 file format first. Users of Access 2000 can open these Access files, and new Access 2002 or later features are available when you use an Access 2000 file in Access 2002 or later. Once you've converted an Access file to Access 2002 - 2003 file format, you can't open the file in Access 2000. However, you can convert an Access 2002 - 2003 file back to Access 2000.

Convert to Access 2002 - 2003 file format only if all users have upgraded to Access 2002 or later, and you have been successfully using the file in Access 2000 file format. If all users have upgraded to Access 2002 or later, you can develop an application in the Access 2002 - 2003 file format without having to make sure that any object, method, property or function you use is also available in Access 2000.

The Access 2002 - 2003 file format offers the following advantages:

- Provision for unknown properties and objects that might exist in future versions of Access
- Ability to save an Access file as an MDE or ADE while in Access 2002 or later
- Improved storage format

 [When to open a previous-version Access database in Access 2002 or later without converting it](#)

You may want to [enable](#) a [Microsoft Access database](#) in a [multiuser](#) environment where all users can't upgrade from Access 97 or earlier to Access 2002 or later at the same time. In this situation, a database must be used simultaneously with different versions of Microsoft Access, and users who have upgraded to Access 2002 or later can enable a database in a previous-version format.

When you're using Access 2002 or later to work with an [enabled](#) previous-version Access database, you can use [objects](#) in the database to view and modify data. However, you can't save changes to database objects. To modify the design of an object or create a new object in the Access database, you must either open it in the previous version or convert the database to Microsoft Access 2000 or Access 2002 - 2003 file format.

 [Converting a previous-version Access file to Access 2000 or Access 2002](#)

[- 2003 file format](#)

You can convert a [Microsoft Access database](#) that was created in Access 2.0 or later to Access 2000 or Access 2002 - 2003 file format. Also, you can convert an [Access project](#) from Access 2000 to Access 2002 -2003 file format.

The following changes might affect the conversion of an Access database:

- **Code compatibility** Access 2000 or later does not support the Microsoft [DAO 2.5/3.x compatibility library](#). If you attempt to convert an Access database in which the code contains older versions of DAO objects, methods, and properties that depend on the DAO 2.5/3.x compatibility library, you receive a message that there were compilation errors during the conversion of the database. Before you convert an Access database, update the code so that it does not depend on the DAO 2.5/3.x compatibility library. If you still receive a message that there were compilation errors during conversion, open the converted database in Access 2002 or later, remove the reference to the missing DAO 2.5/3.x compatibility library, and then set a reference to the Microsoft DAO 3.6 Object Library.

If your Access database uses [add-ins](#) or [library databases](#) created in a Access 97 or earlier, you must convert them as well. However, you can use an add-in or library database created in Access 2000 with a file in Access 2002 - 2003 format.

- **DoMenuItem replaced with RunCommand** The DoMenuItem action is replaced in Microsoft Access 2000 or later with the RunCommand action; DoMenuItem is still supported for backwards compatibility. When you convert a database, the DoMenuItem action is automatically converted to the RunCommand action.
- **Toolbars and menu bars** Access 2002 or later still supports menu bar macros for applications created with previous version of Access. In Access 2000 or later, menu bars and shortcut menus (as well as toolbars) are created and customized from the **Customize** dialog box. You can also use this dialog box to customize the built-in menu bar and toolbars.

Microsoft Access 97 or later supports a new style of toolbars and menu bars. When you convert a Microsoft Access version 2.0 or 95 database to Microsoft Access 2000, any custom toolbars, as well as built-in toolbars,

are automatically converted to the new style of toolbar. Custom menu bars, created with the Microsoft Access 95 Menu Builder or with macros that use AddMenu actions, are interpreted as the new-style menu bars when you open a converted Access database. However, they are not automatically converted, and you can't use the **Customize** dialog box to edit them until you have converted them.

[Converting a previous-version security-enabled Access database to Access 2000 or Access 2002 - 2003 file format](#)

Converting a security-enabled [Microsoft Access project](#) from Access 2000 to Access 2002 - 2003 file format requires no additional action because security is maintained in the [Microsoft SQL Server database](#) that the Access project is connected to. However, there are additional considerations when you convert a security-enabled [Access database](#) from a previous version.

If all users of a previous-version security-enabled database will be upgrading to Microsoft Access 2000 or later, you should convert the Access database to Access 2000 file format. If all users have upgraded to Access 2002 or later, you can convert the Access database to Access 2002 - 2003 file format. If necessary, you can still use the corresponding workgroup information file without converting it. However, recreating the workgroup information file in Access 2002 or later is recommended.

If only some users of a security-enabled database will be upgrading to Access 2002 or later, you can share the database and workgroup information file across all versions of Microsoft Access currently in use.

Important [User-level security](#) in Microsoft Access 2000 or later does not help protect Visual Basic code in [modules](#) or code behind forms and reports. Code in a security-enabled database that you have converted from Access 97 or earlier does not maintain security measures until you create a password for the [Visual Basic project](#).

Before converting a security-enabled [Access database](#), make sure you're using the [workgroup information file](#) that defines the user accounts used to access the database. The user account you use to log on during conversion must have the following permissions:

- Open/Run and Open Exclusive permissions for the database itself
- Administer permission on the MSysACES and MSysObjects system tables
- Modify Design permissions for all other tables in the database, or you must be the owner of all tables in the database
- Read Design permissions for all objects in the database

[Converting a previous-version replica set to Access 2000 or Access 2002 - 2003 file format](#)

- In a [replica](#), you can specify that a table, query or relationship in a replica is local so that it is stored only in that member of the replica set. However, in Access 2000 or later, you can no longer create local forms, reports, macros, modules, or command bars in a converted replica.
- You should synchronize a replica set before converting it.
- If your replica set contains [partial replicas](#), you should synchronize the replica set before conversion and then recreate the partial replicas after conversion.

Caution Synchronizing partial replicas in Access 2000 file format with a [Design Master](#) in Access 2002 - 2003 file format might result in the loss of [database objects](#) in the [Visual Basic for Applications \(VBA\) project](#) of the Access database.

- If you convert a Design Master from Access 95 or 97 to Access 2000 or Access 2002 - 2003 file format, you can no longer synchronize with other Access 95 or 97 replicas. You must convert the Access 95 or 97 replicas to the appropriate file format (Access 2000 or later).

A Design Master in Access 2000 or later can synchronize only with replicas in Access 2000 or Access 2002 - 2003 file format. The only way to convert Access 2000 replicas to Access 2002 - 2003 file format is by synchronizing them with a converted Design Master.

Caution If you convert a Design Master from Access 2000 to Access 2002 - 2003 file format and then synchronize with other replicas in Access 2000 file format, those replicas are converted to Access 2002 - 2003 file format,

and you won't be able to open them in Access 2000.

[Converting an Access 2000 or Access 2002 - 2003 file to a previous version](#)

You can convert a [Microsoft Access database](#) in Access 2000 or Access 2002 - 2003 file format back to Access 97. You can also convert an Access database or an [Access project](#) in Access 2002 - 2003 file format back to Access 2000 file format.

[Converting an Access 2000 or later database back to Access 97](#)

You can convert a Microsoft Access 2000 or later database to an Access 97 database if it is not a member of a [replica set](#). Any functionality that is specific to Access 2000 or later is lost when you convert the database to Access 97. You can't convert an [Access project](#) back to Access 97.

Important Unlike Access 2000 or later, Access 97 is not designed to work with [ActiveX Data Objects \(ADO\)](#).

Converting an Access 2000 or later database to an Access 97 database makes the following changes to the converted database.

- Links to [data access pages](#) are lost. However, the page files (the HTML files that exist outside of the Access database) remain intact.
- Data that relies on [Unicode](#) support in a database in Access 2000 or Access 2002 -2003 file format might not be converted correctly. Access 97 uses characters from only a single code page— a numbered set of 256 characters. For example, an Access 97 database might use only [ANSI](#) characters. During conversion, the collating order of the original Access 2000 database determines which set of characters the resulting Access 97 database can use; any characters that are not in this set are not converted correctly.
- An Access 2000 or later database that contains a table with a **FieldSize** property of **Decimal** cannot be converted to Access 97. You must change the **FieldSize** property to another value such as **Single** or **Double**, or you must change the field's data type to Currency before you can successfully convert the database.

If you are converting a security-enabled Access database, the user account that you use to log on must have the following permissions:

- Open/Run and Open Exclusive permissions for the database itself. Read Design permissions for all objects in the database.
- To help protect the converted database, you must open it in Microsoft Access 97 and apply [user-level security](#). When you convert a database in Access 2000 or Access 2002 - 2003 file format back to Access 97, permissions in the new Access 97 database are reset to their defaults because Access 97 can't use a workgroup information file in Access 2000 or Access 2002 - 2003 file format.

If you have employed a password to help protect the Access database, you can convert it without removing the password. If you have employed a password to help protect the Microsoft Visual Basic for Applications (VBA) code, you must supply the password in the [Visual Basic Editor](#) during conversion.

[Converting an Access 2002 - 2003 file back to Access 2000](#)

You can convert an Access 2002 or later database or Access 2002 or later project back to Access 2000 file format. When you use an Access 2000 file in Access 2002 or later, new Access 2002 or later features are still available. However, those features are not available when you use the same file in Access 2000. If the file contains Visual Basic procedures that use objects, functions, properties, methods, or arguments available only in Access 2002 or later, you'll receive a compile error when you open the Access file in Access 2000.

[The Conversion Errors table](#)

If Microsoft Access encounters errors while converting an [Access file](#), you can view a summary of these errors by opening the Conversion Errors table in the new Access file. The Conversion Errors table contains the following columns:

Object Type The type of [database object](#) in which Access encountered an error, or "Database" if Access encountered an error that is not specific to a particular type of object.

Object Name The name of the object in which Access encountered an error. If Access encounters a compilation error during conversion, however, the name of

the module that contains the error is not specified.

Error Description If necessary, you can press SHIFT+F2 to view the entire description of the error.

 [About opening a previous-version Access file without converting it](#)

When you open a Microsoft Access version 2.0 database, an Access 95 database, or an Access 97 database in Access 2002 or later, the **Convert/Open Database** dialog box asks you to choose between converting and opening (enabling) the Access database. When an Access 2002 or later user enables the database, Access maintains the original format so that users of previous versions of Access can continue to use the database. For example, if a shared database is created in Access version 2.0, it can be used with Access version 2.0 and enabled in Access 2002 or later.

When you use an enabled Access database in Access 2002 or later, you can view [database objects](#) and add, delete, or modify records in Access 2002 or later, but you can't modify the design of objects. To modify the design of existing objects or to add new objects, you must open the file in the previous version of Access.

Size of an enabled Access database Your Microsoft Access database may increase in size when you enable it. If you have already enabled a database in a previous version of Access, and if you enable it again in Access 2002 or later, the size of the Visual Basic [project](#) will increase even more. In rare cases, you may have enabled a Access version 2.0 database in Access 95, enabled it again in Access 97, and enabled it again in Access 2000 or later; in this case, the Visual Basic project must store information in the format of each version.

Enabling a security-enabled Access database If your Microsoft Access database has enabled security, you can make sure that it retains its security by recreating the workgroup information file in Access 2002 or later. If you cannot recreate the workgroup information file, you can use the Workgroup Administrator to join the security-enabled database's previous-version workgroup information file. However, it is recommended that you convert the workgroup information file to Access 2000 file format before you join it. In Access 2002 or later, you can use a workgroup information file that was created in a previous version; however you can't use a workgroup information file created in Access 2002 or later in Access 97 or earlier.

As long as you use the database with its original workgroup information, it retains all its security except permissions for the modules. In Access 2000 or later, you can't use [user-level security](#) to help protect modules. However, you can help protect Visual Basic for Applications (VBA) code with a password.

Code compatibility Microsoft Access 2000 or later does not support the [DAO 2.5/3.x](#) compatibility library. If you attempt to enable an Access database in which the code contains older versions of DAO objects, methods, and properties that depend on the DAO 2.5/3.x compatibility library, you receive a message that there were compilation errors during the enabling of the database. Before you enable the Access database, update the code so that it does not depend on the DAO 2.5/3.x compatibility library.

New style of toolbars and menu bars Microsoft Access 97 or later supports a new style of toolbars and menu bars. When you enable an Access 95 database in Access 2002 or later, custom toolbars are converted to the new style but the conversion isn't saved. Custom menu bars are interpreted as the new style menu bar, but the menu bar macros are not converted and continue to be supported.

[Show All](#)

Convert an Access file

Important You should back up a [Microsoft Access file](#) before you convert it.

 [Convert a previous-version Access file to Microsoft Access 2000 or Access 2002 - 2003 file format](#)

You can convert a [Microsoft Access database](#) from Access 2.0 or later to Access 2000 or Access 2002 - 2003 file format. You can also convert an [Access project](#) from Access 2000 file format to Access 2002 - 2003 file format.

1. If possible, compile the Microsoft Access file you're going to convert— this reduces the possibility of errors during conversion.

You can open and compile an Access 2000 file in Access 2002 or later. However, if the file is an Access database created in Access 97 or earlier, you must do this in the previous version.

 [How?](#)

1. Open a [module](#) in [Design view](#).
2. Do one of the following:
 - In Access 2002 or later, on the **Debug** menu, click **Compile <ProjectName>**.
 - In Access 95 or 97, on the **Debug** menu, click **Compile All Modules**.
 - In Access 2.0, on the **Run** menu, click **Compile Loaded Modules**.
2. Close the Microsoft Access file you're going to convert. If the file is a [multiuser](#) Access database located on a server or in a shared folder, make sure that no one else has it open.

3. In Access 2002 or later, do one of the following:

[Convert an Access database from Access 97 or earlier to the default file format \(Access 2000 or Access 2002 - 2003\)](#)

1. On the **File** menu, click **Open** .
2. Click a shortcut in the left side of the **Open** dialog box, or in the **Look in** box, click the drive or folder that contains the Access file that you want.
3. In the folder list, double-click folders until you open the folder that contains the database. If you can't find the database that you want to open, click **Tools** on the toolbar in the **Open** dialog box, and then click **Search** . In the **Search** dialog box, enter additional search criteria.
4. Double-click the file you want to open.
5. In the **Convert/Open Database** dialog box, click **Convert database**.

[Convert a previous-version Access file to the file format you specify](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.

Notes

- When you convert a previous-version Microsoft Access database that has linked tables, make sure that the external tables are in the same folder that they were originally in. If Microsoft Access can't find the tables that are linked to a database that's being converted, you won't be able to use the converted database. Once you convert the Access database, you can move these tables and use the Linked Table Manager to re-link them. Converting a database that has tables linked to it does not convert the linked tables.
- You can also convert a previous-version Access database by importing its database objects into a new Access 2000 or later database. You can use this method if you don't want all of the objects in the previous-version database to exist in the Access 2000 database. When you convert a database by importing its database objects, you don't automatically import references to libraries, and you might need to reset those references in the new Access file.
- When you convert an Access 2000 file that contains data access pages, Access does not automatically convert the pages. However, when you open a data access page in Design view, Access converts it to use the most recent version of the Microsoft Office Web Components and creates a backup copy of the original page. You can use the backup copy if you want to revert to the Microsoft Office 2000 Web Components.

[Convert an Access file to a previous version](#)

You can convert a [Microsoft Access database](#) in Access 2000 or Access 2002 - 2003 file format back to Access 97. You can also convert an Access database or an [Access project](#) in Access 2002 - 2003 file format back to Access 2000 file format.

1. Open the Microsoft Access file that you want to convert. If it is a [multiuser](#) Access database that is located on a server or in a shared folder, make sure that no one else has it open.

Important If you have employed the security features of the Microsoft Visual Basic for Applications (VBA) code, you must supply the password before you convert the file by clicking **Code**  on the **Database** toolbar and entering the password in the *databasename* **Password** dialog box.

If you are converting a security-enabled Access database, you must have Open/Run and Open Exclusive permissions for the database itself, and Read Design permissions for all objects in the database. To help protect a database that has been converted to Access 97, you must open it in Access 97 and apply [user-level security](#).

2. On the **Tools** menu, point to **Database Utilities**, click **Convert Database/Project**, and then click the file format you want.
3. In the **Convert Database Into** dialog box, type the name of the new previous-version database that you want to create in the **File name** box, and then click **Save**.
4. If the database that you are converting to Access 97 contains code, open the new database in Access 97, open a module and click **References** on the **Tools** menu, and in the **Available References** box, clear the check boxes next to any missing references and set a reference to the Microsoft Access DAO 3.51 Object Library.

If the Access database uses [add-ins](#) or [library databases](#), you must convert them back to Access 97 as well.

[Convert a security-enabled database from Access 95 or 97 to Access 2000 or Access 2002 - 2003 file format](#)

After upgrading from Microsoft Access 95 or 97, you must convert your security-enabled [Access database](#), but you don't need to convert the [workgroup information file](#) to use it with Microsoft Access 2000. However, you should compact the workgroup information file before using it. The following procedure doesn't apply to an Access project, because its security is maintained in the [Microsoft SQL Server database](#) that the project is connected to.

1. Convert the security-enabled Access database.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located

on a server or in a shared folder, make sure that no one else has it open.

4. Start Access.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
2. In Access 2002 or later, open the new Access database, and then compact it by pointing to **Database Utilities** on the **Tools** menu and clicking **Compact and Repair Database**.
3. Close the Access database and temporarily join a different workgroup information information (.mdw) file than the one that is used with the security-enabled database.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

If another workgroup information file doesn't exist, you might have to create one first.

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.
6. Click **OK**.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join

the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

4. Quit and then restart Access.
5. Compact the workgroup information file that was used with the security-enabled database by pointing to **Database Utilities** on the **Tools** menu and clicking **Compact and Repair Database**.
6. Tell users to run Workgroup Administrator and join the compacted workgroup information file before opening the security-enabled database.

[Convert a workgroup information file to Access 2000 or Access 2002 - 2003 file format](#)

You do not need to convert a Microsoft Access 2000 [workgroup information file](#) to Access 2002 or later. However, to take advantage of security and performance improvements, you should re-create workgroup information files from Access 97 or earlier as described below.

1. Create a new workgroup information file, making sure to enter the exact, case-sensitive name, company name, and [workgroup ID](#) that was used to create the original file. Failure to re-enter the exact entries that were used to create the original file will create an invalid [Admins group](#).

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.
6. Click **OK**.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

2. Re-create any group accounts, making sure to enter the exact, case-sensitive group name and [personal ID \(PID\)](#) for each group.

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a

different workgroup to create the database, change your workgroup before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

3. Re-create each user account, making sure to enter the exact, case-sensitive user name and PID for each user.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to

use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
- The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

[Convert a security-enabled database from Access 2.0 to Access 2000 or Access 2002 - 2003 file format](#)

1. Convert the security-enabled database.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.

2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
2. Convert the workgroup information file (typically named system.mda) that is used with the security-enabled database by following the previous procedure.
3. Tell users to run Workgroup Administrator and join the compacted workgroup information file before opening the security-enabled database.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**,

or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

[Convert a previous-version replica set to Access 2000 or Access 2002 - 2003 file format](#)

Take a conservative approach when converting a replica set. It's important that you do not [synchronize](#) the test [Design Master](#) and [replicas](#) you create by following these procedures with members of your working [replica set](#). Otherwise, unintended changes to your data or the structure of your working replica set might occur.

Caution

- If you convert a Design Master in Access 2000 file format to Access 2002 - 2003 file format and then synchronize with other replicas, Access converts the other replicas to Access 2002 - 2003 file format. You will no longer be able to open those replicas in Access 2000.
 - If your replica set contains [partial replicas](#), you should synchronize the replica set before conversion and then recreate the partial replicas after conversion. Synchronizing partial replicas in Access 2000 file format with a [Design Master](#) in Access 2002 - 2003 file format might result in the loss of [database objects](#) in the [VBA project](#) of the Access database.
1. If you are converting to Access 2000 file format, make sure that all users who will access the replica set have installed Access 2000 or later. If you are converting to Access 2002 - 2003 file format, make sure that all users who will access the replica set have installed Access 2002 or later.
 2. Make a copy of the current Design Master that is still in the file format of the earlier version of Access, and put the copy on a different computer or isolate it completely from any other members of the replica set.
 3. In the previous version of Access, make the isolated copy the new Design

Master by pointing to **Replication** on the **Tools** menu and then clicking **Recover Design Master**.

If you are converting from Access 97 or earlier, you must recover the Design Master in the previous version of Access.

4. Create some replicas based on the new Design Master.
5. On the **Tools** menu, point to **Database Utilities**, and then click **Convert Database** to convert the new Design Master to Microsoft Access 2000 or later.

If you are converting from Access 97 or earlier, repeat this step for each new replica in your replica set. Replicas in Access 2000 or Access 2002 - 2003 file format cannot synchronize with Access 95 or Access 97 replicas.

6. Synchronize the new converted Design Master with the new replicas.
7. Run any tests you'd like to try.

If you want to change some objects, you can either make the changes again in the original Design Master after it is converted, or keep the copy around long enough to import the changed objects from it.

8. After you are satisfied that the copy works, delete the new Design Master and all of its replicas.
9. Convert the original Design Master to the newer version of Microsoft Access.

If you are converting from Access 97 or earlier, convert all of the original replicas.

10. Synchronize the replicas.

After you convert a replica set from Microsoft Access 97 or earlier to Access 2000 or later, you can't open the replicas in the previous version of Access.

[Show All](#)

Access 2002 or later features that aren't available in Access 2000

Features that are new in Microsoft Access 2002 or later are available when you use a [Microsoft Access 2000 file](#) in Access 2002 - 2003 file format, but aren't available when you use the same file in Access 2000. These features include the following:

- [PivotTable view](#) and [PivotChart view](#).
- Ability to use queries in [ANSI-92 query mode](#) without using the [ActiveX Data Objects \(ADO\)](#) in a Microsoft Visual Basic [procedure](#) in an [Access database](#).
- Ability to import and export [XML](#).
- Speech and handwriting recognition.
- Multiple undo and redo.
- Binding of reports, list boxes, and combo boxes to ADO [recordsets](#).
- Ability to specify the icon used in the title bar of forms and reports in a particular Access file.
- Adding, editing, or deleting records in an [Access project](#) without defining a [primary key](#) in the table.
- Support for [user-defined functions](#), extended properties, indexed views, and named parameters in an Access project.
- Ability to choose encoding when outputting a database object to text or HTML.
- Increased limit of 32,750 characters for a SQL statement that serves as the **RecordSource** property or **RowSource** property for a form, report, or for a control such as a [list box](#), [combo box](#), [unbound object frame](#), or [Microsoft Office Web Component](#). In Access 2000, this limit is approximately 2000 characters.

In most cases, you can use an Access file that implements an Access 2002 or later feature in Access 2000 without any problems. However, if you open an Access 2000 file in Access 2002 or later and create a Visual Basic procedure that uses an object, function, property, method, or argument that is available only in Access 2002 or later, you'll receive a compile error when you open the Access

file in Access 2000.

[Show All](#)

About using an Access file with multiple versions of Access

In some situations, the users of a shared [Microsoft Access database](#) may not all be able to upgrade to Access 2002 or later at the same time. In these cases, you can share an Access database with users of different versions of Microsoft Access without having to alter your original database.

You can use an Access 2000 database or [Access 2000 project](#) in Access 2002 or later by simply opening the file— no further action is necessary. You can make design changes that take advantage of features new in Access 2002 or later. When you open the file in Access 2000, any functionality specific to Access 2002 or later is simply ignored. You can make design changes in Access 2000. When you open the file in Access 2002 or later, the design changes that take advantage of feature specific to Access 2002 or later remain intact in most cases.

Note An Access file that you use in both Access 2000 and Access 2002 or later can take advantage of features that are specific to Access 2002 or later. However, if the file contains Visual Basic code that uses objects, methods, properties, events, functions, or arguments that are available only in Access 2002 or later, you'll receive a compile error when you try to use the file in Access 2000.

In Access 2002 or later, you can also open a database created in Access 97 or earlier by enabling it.



[Developing an application for use in Access 2000 or later](#)

If you develop an application for use in Access 2000 or later, compiling the application in Access 2000 before distributing it has two advantages:

Better performance in Access 2000. When you open a Microsoft Access 2000 file for the first time in Access 2002 or later and save changes to any [module](#) — including a module containing the code behind a form or report— Access increments the version of the [Visual Basic project](#). This prevents Access 2000 from loading the compiled version of the Visual Basic project. As a result, the application might run more slowly in Access 2000. To ensure optimal

performance of an Access 2000 file that you will use in Access 2000 or later, compile and save the application in Access 2000.

Preventing errors caused by differences between versions. The object libraries that ship with Access 2002 or later contain some new objects, methods, properties, and events that aren't available in Access 2000. Using one of these new objects, methods, properties, or events in Access 2000 will result in a compile error. If you want to use an Access file in Access 2000 or later, compile and save the application in Access 2000 to ensure that the objects, methods, properties, and events in your code are available in both versions.

[Opening a previous-version Access database without converting it](#)

When you open a Microsoft Access version 2.0 database, an Access 95 database, or an Access 97 database in Access 2002 or later, the **Convert/Open Database** dialog box asks you to choose between converting and opening (enabling) the Access database. When an Access 2002 or later user enables the database, Access maintains the original format so that users of previous versions of Access can continue to use the database. For example, if a shared database is created in Access version 2.0, it can be used with Access version 2.0 and enabled in Access 2002 or later.

When you use an enabled Access database in Access 2002 or later, you can view [database objects](#) and add, delete, or modify records in Access 2002 or later, but you can't modify the design of objects. To modify the design of existing objects or to add new objects, you must open the file in the previous version of Access.

Size of an enabled Access database Your Microsoft Access database may increase in size when you enable it. If you have already enabled a database in a previous version of Access, and if you enable it again in Access 2002 or later, the size of the Visual Basic [project](#) will increase even more. In rare cases, you may have enabled a Access version 2.0 database in Access 95, enabled it again in Access 97, and enabled it again in Access 2000 or later; in this case, the Visual Basic project must store information in the format of each version.

Enabling an Access database employing security measures If your Microsoft Access database employs these security measures, you can make sure that it retains its security level by recreating the workgroup information file in Access 2002 or later. If you cannot recreate the workgroup information file, you

can use the Workgroup Administrator to join the security-enabled database's previous-version workgroup information file. However, it is recommended that you convert the workgroup information file to Access 2000 file format before you join it. In Access 2002 or later, you can use a workgroup information file that was created in a previous version; however you can't use a workgroup information file created in Access 2002 or later in Access 97 or earlier.

As long as you use the database with its original workgroup information, it retains all its security measures except permissions for the modules. In Access 2000 or later, you can't use [user-level security](#) to protect modules. However, you can help protect Visual Basic for Applications (VBA) code with a password.

Code compatibility Microsoft Access 2000 or later does not support the [DAO 2.5/3.x compatibility library](#). If you attempt to enable an Access database in which the code contains older versions of DAO objects, methods, and properties that depend on the DAO 2.5/3.x compatibility library, you receive a message that there were compilation errors during the enabling of the database. Before you enable the Access database, update the code so that it does not depend on the DAO 2.5/3.x compatibility library.

New style of toolbars and menu bars Microsoft Access 97 or later supports a new style of toolbars and menu bars. When you enable an Access 95 database in Access 2002 or later, custom toolbars are converted to the new style but the conversion isn't saved. Custom menu bars are interpreted as the new style menu bar, but the menu bar macros are not converted and continue to be supported.



[Using Access tables with several versions of a front-end database](#)

You can upgrade parts of an Access database to Access 2002 or later so that it can take advantage of many new features, and still use the original unchanged database for users of previous versions of Microsoft Access. Users of all versions can share the same data.

You can use this strategy whether your Access database is in one file, or is a [front-end/back-end application](#). To use an Access database that is one file in several versions of Microsoft Access, convert it to a front-end/back-end application. The data in the back end remains in the oldest version of Microsoft Access, and you use a front end that has been converted to Access 2002 or later. To use a front-end/back-end application with several versions of Microsoft

Access, you convert the front end of the application.

[When should I convert or enable a previous-version Access file?](#)

Whether you should convert or enable a [Microsoft Access file](#) depends on which versions of Access users have upgraded to.

[When to convert from Access 97 or earlier to Access 2000 or Access 2002 - 2003 file format](#)

In most cases, you'll want to convert a previous-version [Microsoft Access database](#) to Access 2000 file format if all users have upgraded to Access 2000 or later, and you'll want to convert to Access 2002 - 2003 file format if all users have upgraded to Access 2002 or later. In Access 2002 or later, you can modify data and make design changes to an Access 2000 file. Although you can also [enable](#) an Access 97 or earlier database in Access 2002 or later without converting it, you can change the design of [objects](#) only in the previous version.

Once you've converted an Access database to Microsoft Access 2000 or Access 2002 - 2003 file format, you can't open it in the previous version of Access. You can convert an Access 2000 database back to Access 97. However, you can't convert an Access 2000 database back to Access version 2.0 or 95.

If your Access database is a [multiuser \(shared\) database](#), and all users can't upgrade to Microsoft Access 2000 or later at the same time, you can split the database so that it is a [front-end/back-end application](#). You can then have different versions of the front end connected to the back end, which remains unaltered. Users of Access 2000 or later can use a converted version of the front end, where they can take advantage of new features.

[When to convert an Access 2000 file to Access 2002 - 2003 file format](#)

If you want to save the Access file as an [MDE](#) or [ADE](#) in Access 2002 or later, you must convert the file to Access 2002 - 2003 file format. Otherwise, you don't necessarily need to convert a Microsoft Access 2000 file to the Access 2002 - 2003 file format. If you're converting Access databases from Access 97 or earlier, it's recommended that you convert to Access 2000 file format first. Users of Access 2000 can open these Access files, and new Access 2002 or later features are available when you use an Access 2000 file in Access 2002 or later.

Once you've converted an Access file to Access 2002 - 2003 file format, you can't open the file in Access 2000. However, you can convert an Access 2002 - 2003 file back to Access 2000.

Convert to Access 2002 - 2003 file format only if all users have upgraded to Access 2002 or later, and you have been successfully using the file in Access 2000 file format. If all users have upgraded to Access 2002 or later, you can develop an application in the Access 2002 - 2003 file format without having to make sure that any object, method, property or function you use is also available in Access 2000.

The Access 2002 - 2003 file format offers the following advantages:

- Provision for unknown properties and objects that might exist in future versions of Access
- Ability to save an Access file as an MDE or ADE while in Access 2002 or later
- Improved storage format

 [When to open a previous-version Access database in Access 2002 or later without converting it](#)

You may want to [enable](#) a [Microsoft Access database](#) in a [multiuser](#) environment where all users can't upgrade from Access 97 or earlier to Access 2002 or later at the same time. In this situation, a database must be used simultaneously with different versions of Microsoft Access, and users who have upgraded to Access 2002 or later can enable a database in a previous-version format.

When you're using Access 2002 or later to work with an [enabled](#) previous-version Access database, you can use [objects](#) in the database to view and modify data. However, you can't save changes to database objects. To modify the design of an object or create a new object in the Access database, you must either open it in the previous version or convert the database to Microsoft Access 2000 or Access 2002 - 2003 file format.

 [Linking and importing tables in multiple versions](#)

In Microsoft Access 97 or earlier, you cannot link or import a table from an Access 2000 or later database. However, you can export a table in an Access

2000 or later database to a previous-version Access database. You can also copy and paste data from a table in an Access 2000 or Access 2002 - 2003 file to a table in a previous version of Access.

[Show All](#)

Use an Access file with multiple versions of Access

You can use a [Microsoft Access 2000 database](#) or [Access 2000 project](#) in both Access 2000 or later— even if the file is an Access database employing security features. If you also want to make an Access database available to users of Access 97 or earlier, use one or more of the following procedures.

[Use a one-file Access database with several versions of Microsoft Access](#)

If your [Microsoft Access database](#) is in one file, and you want to use it in several versions of Microsoft Access, convert it to a [front-end/back-end application](#). The data remains in the oldest version of Access, and you use a front end that has been converted to Access 2000 or later.

1. Convert the Access database to Access 2000 or Access 2002 - 2003 file format. When converting, specify a new name for the file.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the

same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
2. On the **Tools** menu, point to **Database Utilities** and use the **Database Splitter** to split the converted database into a front-end/back-end application.
3. Delete the back-end database that the **Database Splitter** created.

You want your data to remain in the original database, so you will use the original database as the back-end database. The back-end database should be in the oldest version of Microsoft Access that is being used.

4. On the **Tools** menu, point to **Database Utilities** and use the **Linked Table Manager** to link the new front-end database to the tables in the previous-version database.

You can then enhance the new front-end database to support new features for users that have upgraded to Microsoft Access 2000 or later. Users of previous versions can continue to use the previous-version database. For example, if the back-end tables are in Microsoft Access version 2.0 format, you can use up several versions of Microsoft Access: Microsoft Access version 2.0 (using the original database), Microsoft Access 95 and 97 (using an enabled version of the original database or a converted front-end database), Microsoft Access 2000 or later (using a converted front-end database).

 [Use a front-end/back-end application with several versions of Microsoft Access](#)

If your [Microsoft Access database](#) is already a [front-end/back-end application](#), you only need to convert the front end to Access 2000 or Access 2002 - 2003 file format.

1. Leave the back-end database alone.
2. Convert the front-end database to Access 2000 or Access 2002 - 2003 file format.

[How?](#)

1. Make a backup copy of the [Microsoft Access file](#) you're going to convert.
2. If possible, compile the Access file in the previous version of Access. This reduces the possibility of errors during conversion.
3. Close the Access file. If the file is a [multiuser Access database](#) located on a server or in a shared folder, make sure that no one else has it open.
4. Start Access.
5. On the **Tools** menu, point to **Database Utilities**, point to **Convert Database**, and then click the file format you want for the new file.
6. In the **Database To Convert From** dialog box, select the database you want to convert and then click **Convert**.
7. In the **Convert Database Into** dialog box, do one of the following:
 - Type a name for the new file.
 - Select a different location for the new file. (You can keep the same name, or change it.)

You cannot convert an Access database into a file with the same name and location as the original database.

8. Click **Save**.
3. On the **Tools** menu, point to **Database Utilities** and use the **Linked Table Manager** to link the new Access 2000 or later front-end database to the tables in the previous-version back-end database. You can then enhance the Access 2000 or later front-end database to support new features for users that have upgraded to Access 2000 or later.

[Open a database created in Access 97 or earlier without converting it](#)

You can open a [Microsoft Access database](#) created in Access 97 or earlier without converting it.

1. Make sure that the Access database you're going to [enable](#) is closed. If the file is a [multiuser](#) Access database located on a server or in a shared folder, make sure that no one else has it open.
2. Click **Open**  on the **Database** toolbar.
3. In the **Open** dialog box, click the previous-version Access database you

want to enable and click **Open**.

4. In the **Convert/Open Database** dialog box, click **Open Database**.

Microsoft Access [enables](#) the previous-version database by storing additional information in the file to accommodate the default file format specified on the **Advanced** tab of the **Options** dialog box. An enabled database can still be opened with its original version of Microsoft Access.

[Use an Access database employing security features with multiple versions of Access](#)

With one exception, the issues involved when sharing a database employing security features across more than one version of Microsoft Access are the same as the issues for sharing an unsecured database across more than one version. The one exception concerns how to handle the [workgroup information files](#) that are used with the security-enabled database.

- Do one of the following:
 - If you want to share a security-enabled database from Access 95 or 97, tell users who will be upgrading to Access 2002 or later to join the appropriate workgroup information file with the oldest version of Access that will be sharing the security-enabled database.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and

then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

Access 2002 or later can use workgroup information files that have been created with any previous version. With one exception, a previous version of Access can't use a workgroup information that was created in a later version. The exception is Access 2000, which can use a workgroup information file that was created in Access 2002 or later if that workgroup information file is in Access 2000 file format.

Important If users will be sharing a security-enabled database from Microsoft Access 95 or 97, you should compact the current workgroup information file with Access 2002 or later before using it. Compacting the file by using Access 2002 or later does not change the file format, so the file can continue to be used by any Microsoft Access 95 or 97 users who are not upgrading.

- If the shared database is Microsoft Access version 2.0, convert the workgroup information file that will be used with the security-enabled database.

[How?](#)

You do not need to convert a Microsoft Access 2000 [workgroup information file](#) to Access 2002 or later. However, to take advantage of security and performance improvements, you should re-create workgroup information files from Access 97 or earlier as described below.

1. Create a new workgroup information file, making sure to enter the exact, case-sensitive name, company name, and [workgroup ID](#) that was used to create the original file. Failure to re-enter the exact entries that were used to create the original file will create an invalid [Admins group](#).

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.
6. Click **OK**.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

2. Re-create any group accounts, making sure to enter the exact, case-sensitive group name and [personal ID \(PID\)](#) for each group.

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

3. Re-create each user account, making sure to enter the exact, case-sensitive user name and PID for each user.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts.

2. Open a database.

3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.

4. On the **Users** tab, click **New**.

5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a

different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
- The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate a encrypted identifier for the [user account](#).

Tell only users who are upgrading to Microsoft Access 2000 or later to join the converted workgroup information file. All users who are not upgrading from version 2.0 must continue to use the workgroup information file produced with their original version.

[Show All](#)

About the Conversion Errors table

If Microsoft Access encounters errors while converting an [Access file](#), you can view a summary of these errors by opening the Conversion Errors table in the new Access file. The Conversion Errors table contains the following columns:

Object Type The type of [database object](#) in which Access encountered an error, or "Database" if Access encountered an error that is not specific to a particular type of object.

Object Name The name of the object in which Access encountered an error. If Access encounters a compilation error during conversion, however, the name of the module that contains the error is not specified.

Error Description If necessary, you can press SHIFT+F2 to view the entire description of the error.

[Show All](#)

Troubleshoot using an Access 2000 file with multiple versions of Access

[I get a compile error when I open a Microsoft Access 2000 file in Access 2000.](#)

Your [Microsoft Access file](#) contains a [Visual Basic procedure](#) that uses an object, [method](#), [property](#), or [event](#) that is available in Access 2002 or later, but is not available in Access 2000. If you want to use an Access file in both Access 2000 and Access 2002 - 2003, compile and save the application in Access 2000 to ensure that the objects, methods, properties and events in your code are available in both versions.

[An Access 2000 application runs more slowly after I save changes in Access 2002 or later.](#)

When you open a [Microsoft Access 2000 file](#) for the first time in Access 2002 or later and save changes to a form, report, macro, or module, Access increments the version of the Visual Basic project. This prevents Access 2000 from loading the compiled version of the Visual Basic project. As a result, the application might run more slowly in Access 2000. To ensure optimal performance of an Access 2000 file that you will use in Access 2000 or later, compile and save the application in Access 2000.

[After synchronizing an Access 2000 replica, I can't open it in Access 2000.](#)

A member of the replica set was converted to Access 2002 - 2003 file format. Synchronizing an Access 2000 database with a database in Access 2002 or later converts the database.

[A form, report, list box, or combo box in an Access 2000 file shows the correct data in Access 2002 or later, but not in Access 2000.](#)

In Access 2002 or later, a SQL statement that serves as the **RecordSource**

property or the **RowSource** property for a form, report, or for a control such as a [list box](#), [combo box](#), [unbound object frame](#), or [Microsoft Office Web Component](#) can contain a maximum of 32,750 characters. In Access 2000, this limit is approximately 2,000 characters.

[A macro in an Access 2000 file that I am using in Access 2000 or later is not working as expected.](#)

Some macro actions that are available in Access 2002 or later might not be available in Microsoft Access 2000. When you open the macro in Design view in Access 2000, a placeholder for any action that is not available appears in the Comments column with the following text:

Macro action from newer Access version: *macro action*

If you want to edit the macro in Access 2000, you can modify actions that come after any instance of placeholder text, and you can insert new actions after the last instance of placeholder text. However, if you modify actions that come before placeholder text, insert new actions before placeholder text, or modify the placeholder itself, the macro might not function as expected in Access 2002 or later.

[I can't see some of the queries in an Access 2000 database when I open it in Access 2000.](#)

In Microsoft Access 2000, queries based on ANSI-92 [SQL query syntax](#) are not visible in the Database window. If you want all of the queries in a Microsoft Access database to be available in Access 2000 or later, open the Access database in Access 2002 or later, click **Options** on the **Tools** menu, click the **Tables/Queries** tab, and under **SQL Server Compatible Syntax (ANSI 92)** clear the **This database** check box. After clearing the check box, you might have to modify some of your queries so they use Microsoft Jet SQL syntax.

[After I open an Access 2000 file in Access 2002 or later, I can't use the data access pages in Access 2000.](#)

In Access 2002 or later, when you open a data access page in [Design view](#), Access makes a backup copy of the page and converts it so that it uses the latest version of the [Microsoft Office Web Components](#). In Access 2000, you can no

longer open the page in Design view, but you can open the backup copy of the page in Access 2000.



[I can't save an Access 2000 file as an MDE or ADE in Access 2002 or later.](#)

When you save a [Microsoft Access file](#) as an [MDE](#) or [ADE](#) file, Access 2002 or later adds information to the file. Because Access 2000 can't interpret this information, you can't save a file in Access 2000 file format as an MDE or ADE file while you are in Access 2002 or later.

To save a file in Access 2000 file format as an MDE or ADE file, do one of the following:

- Open the Access file in Access 2000, and then save it as an MDE or ADE file.
- In Access 2002 or later, convert the Access file to Access 2002 - 2003 file format, and then save it as an MDE or ADE file.

[Show All](#)

Troubleshoot tables and field properties (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[I can't save a table after I set a primary key.](#)

Microsoft Access can't create a [primary key](#) for a field that already contains data if that data includes duplicate values or [Null](#) values. If you can't readily eliminate **Null** values or duplicate entries by editing your data, do one of the following:

- To locate records with duplicate values, use a Find Duplicates query.
- To locate records with **Null** values, create a select query, and type the expression **Is Null** in the **Criteria** cell for the field where you want to set the primary key.
- Choose a different field for your primary key.
- Add an [AutoNumber](#) field and set it as the primary key.
- Define a multiple-field primary key.

[The caption of a field in my table has been deleted.](#)

If you rename a field with a caption while in [Datasheet view](#), the caption is deleted and replaced with the new field name. To redefine the field's caption, open the table in [Design view](#), click the field for which you want to define a different column name, and in the lower portion of the window, type the new column name in the **Caption** property box.

[When I compact my Microsoft Access database after changing the starting value of a AutoNumber field, the AutoNumber value for the new record is reset to 1.](#)

If you want to compact the database after changing the starting [AutoNumber](#) value, make sure to add at least one record to the table first. If you don't, when

you compact the database, the AutoNumber value for the next record added will be reset to 1 more than the highest previous value. For example, if there were no records in the table when you reset the starting value, compacting would set the AutoNumber value for the next record added to 1; for example, if there were records in the table when you reset the starting value and the highest previous value was 50, compacting would set the AutoNumber value for the next record added to 51.

[When I type a number in a field or control, only zeros appear to the right of the decimal point.](#)

The **FieldSize** property of the field is **Integer** or **Long Integer**. Integers are whole numbers that contain no decimals or fractions. Even when the **DecimalPlaces** property of an integer field is set to a number greater than zero, only zeros appear to the right of the decimal point. Change the **FieldSize** property, or change the [data type](#) to Currency.

[Show All](#)

About tables (MDB)

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means that you store that data only once. This results in a more efficient database and fewer data-entry errors.

 [How data is organized in tables](#)

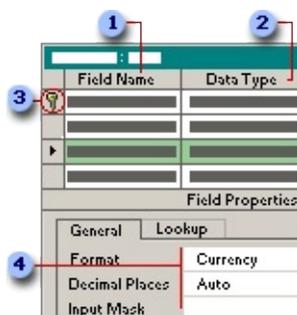
Tables organize data into columns (called fields) and rows (called records).

Products : Table		
Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camaron Tiger	2	53

For example, each field in a Products table contains the same type of information for every product, such as the product's name. Each record in that table contains all the information about one product, such as the product's name, supplier ID number, units in stock, and so on.

 [Table Design view](#)

In table Design view, you can create an entire table from scratch, or add, delete, or customize the fields in an existing table.



1 If you want to track additional data in a table, add more fields. If an existing field name isn't descriptive enough, you can rename the field.

- 2 Setting a field's [data type](#) defines what kind of values you can enter in a field. For example, if you want a field to store numerical values that you can use in calculations, set its data type to **Number** or **Currency**.
- 3 You use a unique tag, called a [primary key](#), to identify each record in your table. A table's primary key is used to refer to related records in other tables.
- 4 Field properties are a set of characteristics that provide additional control over how the data in a field is stored, entered, or displayed. Which properties are available depends on a field's data type.

[How to relate two tables](#)

A common field relates two tables so that Microsoft Access can bring together the data from the two tables for viewing, editing, or printing. In one table, the field is a [primary key](#) that you set in table [Design view](#). That same field also exists in the related table as a [foreign key](#).

Suppliers : Table	
Supplier ID	Company Name
1	Exotic Liquid
2	New Orleans
3	Grandma Kell
4	Tokyo Trade

Products : Table		
Product Name	Supplier ID	Units in Stock
Chai	1	39
Chang	1	17
Aniseed Syrup	1	13
Camarvon Tiger	2	53

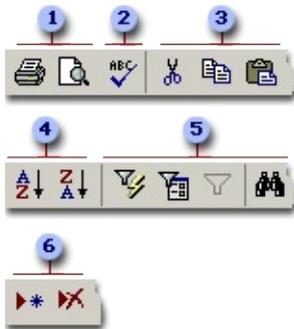
- 1 In the Suppliers table, you enter a supplier ID, company name, and so on for each supplier. SupplierID is the primary key that you set in table Design view.
- 2 In the Products table, you include the SupplierID field, so that when you enter a new product, you can identify its supplier by entering that supplier's unique ID number. SupplierID is the foreign key in the Products table.

[Table Datasheet view](#)

In a table or query, [Datasheet view](#) provides the tools you need to work with data.

[Using the Table Datasheet and Query Datasheet toolbars](#)

The **Table Datasheet** and **Query Datasheet toolbars** provide many of the tools you need to find, edit, and print records.



- 1 Print or preview data.
- 2 Check spelling.
- 3 Cut, copy, or paste selected text, fields, whole records, or the entire datasheet.
- 4 Sort records.
- 5 Filter records, and find or replace values.
- 6 Add or delete records.

[Working with columns, rows, and subdatasheets](#)

You can find tools for working with columns, rows, and [subdatasheets](#) in the datasheet itself, or by right-clicking a [column selector](#).

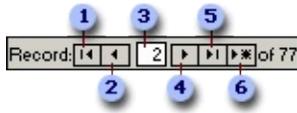


- 1 Use the column selector to move, hide, or rename a column.
- 2 Resize columns or rows.
- 3 Use subdatasheets to view related data.

- 4 Freeze the leftmost column so that it is displayed as you scroll to the right.

[Moving through records](#)

You can use the navigation toolbar to move through the records in a datasheet.



- 1 Go to the first record.
- 2 Go to the previous record.
- 3 Type a record number to move to.
- 4 Go to the next record.
- 5 Go to the last record.
- 6 Go to a blank (new) record.

[Show All](#)

Open a database object

[Open a table](#)

1. In the [Database window](#), click **Tables**  under **Objects**.
2. Click the table you want to open.
3. Do one of the following:

To open the table in [Design view](#), click **Design** on the Database window toolbar.

To open the table in [Datasheet view](#), click **Open** on the Database window toolbar.

[Open a form or subform](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the form or subform you want to open.
3. Do one of the following:

To open the form in [Design view](#), click **Design** on the Database window toolbar.

To open the form in [Form view](#), click **Open** on the Database window toolbar.

Note If the main form is open in Design view, and you want to open a subform in a separate window, select the subform, and click **Subform in New Window** on the **View** menu. The subform control in the main form will turn white and display the name of the subform. You must save changes to the main form and subform individually.

[Open a report or subreport](#)

1. In the [Database window](#), click **Reports**  under **Objects**.
2. Click the report or subreport you want to open.

3. Do one of the following:

To open the report in [Design view](#), click **Design** on the Database window toolbar.

To open the report in [Print Preview](#), click **Preview** on the Database window toolbar.

[Open a query \(select or crosstab\) that shows records](#)

1. In the [Database window](#), click **Queries**  under **Objects**.
2. Click the query you want to open.
3. Do one of the following:

To open the query in [Design view](#), click **Design** on the Database window toolbar.

To open the query in [Datasheet view](#), click **Open** on the Database window toolbar.

Notes

- When you open a select or crosstab query in Datasheet view, you are actually executing the query.
- If you previously saved a query while it was open in [SQL view](#), Microsoft Access opens the query in SQL view the next time you click **Design** on the Database window toolbar for that query.

[Open a data access page](#)

To open a [data access page](#), you must have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later on your computer.

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Click the data access page you want to open.
3. Do one of the following:

To open the page in [Design view](#), click **Design** on the Database window toolbar.

To open the page in [Page view](#), click **Open** on the Database window toolbar.

To open the page in Internet Explorer, do one of the following:

- Open the page in Design view or Page view, and then click **Web Page Preview** on the **File** menu.
- If Internet Explorer is your default Web browser, in Windows Explorer, double-click the HTML file to open the page in Internet Explorer.

Note You can also open a page in Page view by clicking **Open** on the **File** menu in Microsoft Access. Access opens the page without opening the database that the page is connected to, so wizards, builders, and add-ins are not available.

[Open a macro](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the name of the macro you want to open.
3. Click **Design** on the Database window toolbar.

[Open a module](#)

1. In the [Database window](#), click **Modules**  under **Objects**.
2. Do one of the following:

To open a new [standard module](#), and then click the **New** button on the Database window toolbar.

To open an existing standard or class module, select the module you want to open, and then click **Design** on the Database window toolbar.

To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the toolbar.

To open a new [class module](#) that isn't associated with a form or report, click **Class Module** on the **Insert** menu.

[Show All](#)

Switch between views of a database object

[Switch between views of a form, report, query, or table](#)

- Click the **View** button on the toolbar. To change to the view indicated by the graphic, click the button itself. To see a list of other views to choose from, click the arrow next to the button.

Note The views displayed in the **View** button list and on the **View** menu depend on the setting of the **AllowFormView**, **AllowDatasheetView**, **AllowPivotTableView**, and **AllowPivotChartView** properties of the form.

[Switch between views of a subform](#)

When a main form is open in Form view, you can switch between the views of its subforms.

1. Open the main form in Form view.
2. Click the subform whose view you want to change.
3. On the **View** menu, point to **Subform**, and click the view you want.

Note The views available for a subform depend on the setting of the **AllowFormView**, **AllowDatasheetView**, **AllowPivotTableView**, and **AllowPivotChartView** properties of the subform.

[Switch between views of a data access page](#)

To open a data access page in [Design view](#), [Page view](#), or in Microsoft Internet Explorer, you must have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later on your computer.

- To switch to another view from within Microsoft Access, click the **View** button on the toolbar. To change to the view indicated by the graphic, click the button itself. To see a list of other views to choose from, click the arrow

next to the button.

- To switch from Internet Explorer to Microsoft Access, click **Edit with Microsoft Access** on the **File** menu.

[Show All](#)

About creating a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To create a blank (empty) table for entering your own data, you can:

- Use the Table Wizard to choose the [fields](#) for your table from a variety of predefined tables such as business contacts, household inventory, or medical records.
- Create a table in [Design view](#), where you can add fields, define how each field appears or handles data, and create a [primary key](#).
- Enter data directly into a blank [datasheet](#). When you save the new datasheet, Microsoft Access will analyze your data and automatically assign the appropriate [data type](#) and [format](#) for each field.

To create a table from existing data, you can:

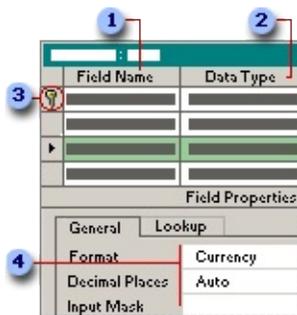
- [Import](#) or link data from another Access database or data in a variety of file formats from other programs.
- Perform a make-table query to create a table based on data in a current table. For example, you can use make-table queries to archive old records, to make backup copies of your tables, to select a group of records to export to another database, or to use as a basis for reports that display data from a specific time.

[Show All](#)

About customizing a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In table Design view, you can create an entire table from scratch, or add, delete, or customize the fields in an existing table.



- 1 If you want to track additional data in a table, add more fields. If an existing field name isn't descriptive enough, you can rename the field.
- 2 Setting a field's [data type](#) defines what kind of values you can enter in a field. For example, if you want a field to store numerical values that you can use in calculations, set its data type to **Number** or **Currency**.
- 3 You use a unique tag, called a [primary key](#), to identify each record in your table. A table's primary key is used to refer to related records in other tables.
- 4 Field properties are a set of characteristics that provide additional control over how the data in a field is stored, entered, or displayed. Which properties are available depends on a field's data type.

[Show All](#)

Create a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Create a table by using the Table Wizard](#)

1. Press F11 to switch to the [Database window](#).
2. Click **Tables**  under **Objects**, and then click **New** on the Database window toolbar.
3. Double-click **Table Wizard**.
4. Follow the directions in the **Table Wizard** dialog boxes.

Note If you want to modify or extend the resulting table, you can do so in [Design view](#) when you have finished using the Table Wizard.

[Create a table in Design view](#)

1. Press F11 to switch to the [Database window](#).
2. Click **Tables**  under **Objects**, and then click **New** on the Database window toolbar.
3. Double-click **Design View**.
4. Define each of the fields in your table.

[How?](#)

1. Open the table in [Design view](#).
2. To insert the field within the table, click in the row below where you want to add the field, and then click **Insert Rows**  on the toolbar.

To add the field to the end of the table, click in the first blank row.

3. Click in the **Field Name** column and type a unique name for the field.
4. In the **Data Type** column, keep the default (**Text**); or click in the **Data Type** column, click the arrow, and select the [data type](#) you want.
5. In the **Description** column, type a description of the information this

field will contain. This description is displayed on the [status bar](#) when adding data to the field and is included in the Object Definition of the table. The description is optional.

5. Define a [primary key](#) field before saving your table.

[How?](#)

1. Open a table in [Design view](#).
2. Select the field or fields you want to define as the [primary key](#).

To select one field, click the [row selector](#) for the desired field.

To select multiple fields, hold down the CTRL key and then click the row selector for each field.

3. Click **Primary Key**  on the toolbar.

Note If you want the order of the fields in a multiple-field primary key to be different from the order of those fields in the table, click **Indexes**  on the toolbar to display the [Indexes window](#), and then reorder the field names for the [index](#) named PrimaryKey.

Note You don't have to define a primary key, but it's usually a good idea. If you don't define a primary key, Microsoft Access asks if you want Access to create one for you when you save the table.

6. When you are ready to save your table, click **Save**  on the toolbar, and then type a unique name for the table.

[Create a table by entering data in a datasheet](#)

1. Press F11 to switch to the [Database window](#).
2. Click **Tables**  under **Objects**, and then click **New** on the Database window toolbar.
3. Double-click **Datasheet View**. A blank datasheet is displayed. The default column names are Field1, Field2, and so on.
4. Rename each column you will use: double-click the column name, type a name for the column, and then press ENTER.

5. You can insert additional columns at any time: click in the column to the right of where you want to insert a new column, and then on the **Insert** menu, click **Column**. Rename the column as described in step 4.
6. Enter your data in the datasheet.

Enter each kind of data in its own column (each column is called a [field](#) in Microsoft Access). For example, if you are entering names, enter the first name in its own column and the last name in a separate column. If you are entering dates, times, or numbers, enter them in a consistent format so that Microsoft Access can create an appropriate [data type](#) and [display format](#) for the column. Any columns you leave empty will be deleted when you save the datasheet.

7. When you've added data to all the columns you want to use, click **Save**  on the toolbar to save your datasheet.
8. Microsoft Access asks you if you want to create a [primary key](#). If you haven't entered data that can be used to uniquely identify each row in your table, such as part numbers or ID numbers, it's recommended that you click **Yes**. If you have entered data that can uniquely identify each row, click **No**, and then specify the field that contains that data as your primary key in [Design view](#).

[How?](#)

1. Open a table in [Design view](#).
2. Select the field or fields you want to define as the [primary key](#).

To select one field, click the [row selector](#) for the desired field.

To select multiple fields, hold down the CTRL key and then click the row selector for each field.

3. Click **Primary Key**  on the toolbar.

Note If you want the order of the fields in a multiple-field primary key to be different from the order of those fields in the table, click **Indexes**  on the toolbar to display the [Indexes window](#), and then reorder the field names for the [index](#) named PrimaryKey.

Microsoft Access will assign data types to each field (column) based on the kind of data you entered. If you want to customize a field's definition further— for example, to change its data type, or define a [validation rule](#), use Design view.

[Show All](#)

About the Table Analyzer's query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The screenshot illustrates the Table Analyzer Wizard process. It shows the original table, derived tables, and the resulting query. The original table has four columns: Product Name, Supplier ID, Company Name, and Address. The derived tables are Products and Suppliers. The query made by the wizard is a join of the original table and the Suppliers table.

Original Table 1			
Product Name	Supplier ID	Company Name	Address
Ravioli Angelo	PAST	Pasta Buttini	Via dei
Carnarvon Tigers	PAVL	Pavlova, Ltd.	74 Ros
Vegie-spread			

Products	
Ravioli Angelo	PAST
Carnarvon Tigers	PAVL
Vegie-spread	

Suppliers			
Supplier ID	Company Name	Address	
PAST	Pasta Buttini	Via dei	
PAVL	Pavlova, Ltd.	74 Ros	

Query Made by the Table Analyzer Wizard			
Ravioli Angelo	PAST, Pasta Buttini		
Carnarvon Tigers	PAVL, Pavlova, Ltd.		
Vegie-spread	PAVL, Pavlova, Ltd.		

- 1 Original table
- 2 Tables created by the Table Analyzer
- 3 Query created by the Table Analyzer
- 4 Lookup list

If your [Microsoft Access database](#) has a table that contains repeating information in one or more fields, you can use the Table Analyzer to split the data into related tables so that you can store data more efficiently. This process is called normalization.

The Table Analyzer divides one table that contains repeating information into separate tables in which each type of information is stored only once. This makes the database more efficient and easier to update, and reduces its size. After the wizard divides the data, you can still view and work with the data in one place by having the wizard create a query.

You can use the query to update data from more than one table at the same time. The query also offers other timesaving features to increase the accuracy of your data.

- If you change a field that is repeated in the query, all records with that value are automatically updated because you are actually updating just one field in the query's underlying table.
- From the new tables, the query inherits Lookup fields, which let you change field values by picking values from a list instead of having to correctly type a value. The Lookup list looks up values from the underlying table. When the same value applies to more than one record, you can ensure accuracy and consistency by picking that value from the list each time.
- Forms, reports, and data access pages that were previously based on the original table will automatically be based on the query now, because the wizard assigns the original table's name to the query and renames the table.
- Existing forms and reports based on the query will inherit automatic field updates. New forms and reports will also inherit Lookup lists.

Convert a linked table to a local table

1. In the **Database** window, select the linked table that you want to convert to a local table.
2. On the **Edit** menu, click **Copy**.
3. On the **Edit** menu, click **Paste**.
4. Type the name for the new table in the **Table Name** box.
5. Click **Structure Only (Local Table)** if you want to create an empty table based on the structure of the linked table. Click **Structure and Data (Local Table)** if you want the new table to contain the data and structure of the linked table.
6. Click **OK**.

[Show All](#)

Split a table into related tables (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. On the **Tools** menu, point to **Analyze**, and then click **Table**.
2. Follow the instructions in the steps of the Table Analyzer Wizard.

Note In a [Microsoft Access project](#), the tables reside in a Microsoft SQL Server database; therefore, the Table Analyzer is not available.

[Show All](#)

Add a field to a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Add a field to a table in Design view](#)

1. Open the table in [Design view](#).
2. To insert the field within the table, click in the row below where you want to add the field, and then click **Insert Rows**  on the toolbar.

To add the field to the end of the table, click in the first blank row.

3. Click in the **Field Name** column and type a unique name for the field.
4. In the **Data Type** column, keep the default (**Text**); or click in the **Data Type** column, click the arrow, and select the [data type](#) you want.
5. In the **Description** column, type a description of the information this field will contain. This description is displayed on the [status bar](#) when adding data to the field and is included in the Object Definition of the table. The description is optional.

[Add a field \(column\) to a table in Datasheet view](#)

1. Open the table in [Datasheet view](#).
2. Click in the column to the right of where you want to insert a new column, and then on the **Insert** menu, click **Column**.
3. Double-click the new column's name, and then type a unique name for the column.

If you want to customize a field's definition further—for example, to change its [data type](#), or define a [validation rule](#), use table [Design view](#).

Note To add a field to a linked table in another [Microsoft Access database](#), you must open its source database. If the linked table is from another application, you must open the source file with that application to add a field.

[Show All](#)

Guidelines for naming fields, controls, and objects

Names of fields, [controls](#), and objects in Microsoft Access:

- Can be up to 64 characters long.
- Can include any combination of letters, numbers, spaces, and special characters except a period (.), an exclamation point (!), an accent grave (`), and brackets ([]).
- Can't begin with leading spaces.
- Can't include control characters (ASCII values 0 through 31).
- Can't include a double quotation mark (") in table, view, or [stored procedure](#) names in a [Microsoft Access project](#).

Although you can include spaces in field, control, and object names, most examples in the Microsoft Access documentation show field and control names without spaces because spaces in names can produce naming conflicts in [Microsoft Visual Basic for Applications](#) in some circumstances.

When you name a field, control, or object, it's a good idea to make sure the name doesn't duplicate the name of a [property](#) or other element used by Microsoft Access; otherwise, your database can produce unexpected behavior in some circumstances. For example, if you refer to the value of a field called Name in a table NameInfo using the syntax NameInfo.Name, Microsoft Access displays the value of the table's **Name** property rather than the value of the Name field.

Another way to avoid unexpected results is to always use the ! operator instead of the . (dot) operator to refer to the value of a field, control, or object. For example, the following identifier explicitly refers to the value of the Name field rather than the **Name** property:

```
[NameInfo]![Name]
```

[Show All](#)

Copy a field and its properties (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

This procedure only copies a field's definition; it doesn't copy any values in the field.

1. Open a table in [Design view](#).
2. Select the field or fields you want to copy.

To select one field, click that field's [row selector](#).

To select a group of fields, drag through the row selectors of those fields.



1 Row selectors

3. Click **Copy**  on the toolbar and then click in the first blank row.
4. Click **Paste**  on the toolbar.
5. Type a new name for the copied field.

[Show All](#)

Delete a field from a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When you delete a field, remember that if other [database objects](#) contain references to the deleted field, you need to delete those references as well. For example, if a report includes a control [bound](#) to the deleted field, Microsoft Access won't be able to find the data from the field and will generate a message.

You can't delete a field that's part of a [relationship](#)— you must delete the relationship first.

[Delete a field from a table in Design view](#)

1. Open the table in [Design view](#).
2. Select the field or fields you want to delete.

To select one field, click that field's [row selector](#).

To select a group of fields, drag through the row selectors of those fields.



1 Row selectors

3. Click **Delete Row**  on the toolbar.

[Delete a field from a table in Datasheet view](#)

1. Open the table in [Datasheet view](#).
2. Click the [field selector](#) of the column you want to delete.

↓ Last Name:
Davolio
Fuller
Leverling

3. On the **Edit** menu, click **Delete Column**.

[Show All](#)

About data types and field sizes (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Decide what kind of [data type](#) to use for a field based on these considerations:

- What kind of values do you want to allow in the field? For example, you can't store text in a field with a Number data type.
- How much storage space do you want to use for values in the field?
- What types of operations do you want to perform on the values in the field? For example, Microsoft Access can sum values in Number or Currency fields, but not values in Text or [OLE Object](#) fields.
- Do you want to sort or [index](#) a field? OLE Object fields can't be sorted or indexed.
- Do you want to use a field to group records in queries or reports? OLE Object fields can't be used to group records.
- How do you want to sort values in a field? In a Text field, numbers sort as strings of characters (1, 10, 100, 2, 20, 200, and so on), not as numeric values. Use a Number or Currency field to sort numbers as numeric values. Also, many date formats will not sort properly if entered in a Text field. Use a Date/Time field to ensure proper sorting for dates.
- Will you need to store Microsoft Word or Microsoft Excel documents, pictures, sound, and other types of binary data created in other programs? OLE objects can be [linked](#) to or [embedded](#) in an OLE Object field in a Microsoft Access table. To display the OLE object, use a [control](#) in a form or report.

[Choosing between a Text or Memo field](#)

Microsoft Access provides two [field data types](#) to store data with text or combinations of text and numbers: Text and Memo.

Use a Text data type to store data such as names, addresses, and any numbers

that do not require calculations, such as phone numbers, part numbers, or postal codes. A Text field can store up to 255 characters, but the default field size is 50 characters. The **FieldSize** property controls the maximum number of characters that can be entered in a Text field.

Use the Memo data type if you need to store more than 255 characters. A Memo field can store up to 65,536 characters. If you want to store formatted text or long documents, you should create an OLE Object field instead of a Memo field.

Both Text and Memo data types store only the characters entered in a field; space characters for unused positions in the field aren't stored.

You can sort or group on a Text field or a Memo field, but Access only uses the first 255 characters when you sort or group on a Memo field.



[Choosing between a Number or Currency field](#)

Microsoft Access provides two [field data types](#) to store data containing numeric values: Number and Currency.

Use a Number field to store numeric data to be used for mathematical calculations, except calculations that involve money or that require a high degree of accuracy. The kind and size of numeric values that can be stored in a Number field is controlled by setting the **FieldSize** property. For example, the Byte field size will only store whole numbers (no decimal values) from 0 to 255 and occupies 1 byte of disk space.

Use a Currency field to prevent rounding off during calculations. A Currency field is accurate to 15 digits to the left of the decimal point and 4 digits to the right. A Currency field occupies 8 bytes of disk space.

Number and Currency fields provide predefined [display formats](#), or you can create a custom format.



[Choosing between an incremented or random AutoNumber field](#)

Microsoft Access provides the AutoNumber data type to create fields that automatically enter a unique number when a record is added. Once a number is generated for a record, it can't be deleted or changed. An AutoNumber field can

generate three kinds of numbers: sequential numbers that increment by one, random numbers, and Replication ID (also referred to as GUIDs— globally unique identifiers) numbers. AutoNumbers that increment by one are the most common kind of AutoNumber and are a good choice for use as a table's [primary key](#). Random AutoNumbers will generate a random number that is unique to each record within the table. Replication ID AutoNumbers are used in [database replication](#) to generate unique identifiers for synchronizing [replicas](#).

[Show All](#)

Field data types available in Access (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The following list summarizes all the [field data types](#) available in Microsoft Access, their uses, and their storage sizes.

Text

Use for text or combinations of text and numbers, such as addresses, or for numbers that do not require calculations, such as phone numbers, part numbers, or postal codes.

Stores up to 255 characters. The **FieldSize** property controls the maximum number of characters that can be entered.

Memo

Use for lengthy text and numbers, such as notes or descriptions.

Stores up to 65,536 characters.

Number

Use for data to be included in mathematical calculations, except calculations involving money (use Currency type).

Stores 1, 2, 4, or 8 bytes; stores 16 bytes for Replication ID (GUID). The **FieldSize** property defines the specific Number type.

Date/Time

Use for dates and times.

Stores 8 bytes.

Currency

Use for currency values and to prevent rounding off during calculations.

Stores 8 bytes.

AutoNumber

Use for unique sequential (incrementing by 1) or random numbers that are automatically inserted when a record is added.

Stores 4 bytes; stores 16 bytes for Replication ID (GUID).

Yes/No

Use for data that can be only one of two possible values, such as Yes/No,

True/False, On/Off. [Null](#) values are not allowed.

Stores 1 bit.

OLE Object

Use for OLE objects (such as Microsoft Word documents, Microsoft Excel spreadsheets, pictures, sounds, or other binary data) that were created in other programs using the [OLE](#) protocol.

Stores up to 1 gigabyte (limited by disk space).

Hyperlink

Use for [hyperlinks](#). A hyperlink can be a [UNC path](#) or a [URL](#).

Stores up to 64,000 characters.

Lookup Wizard

Use to create a field that allows you to choose a value from another table or from a list of values using a combo box— -choosing this option in the data type list starts a wizard to define this for you.

Requires the same storage size as the [primary key](#) that corresponds to the [Lookup field](#)— -typically 4 bytes.

[Show All](#)

About changing a field's data type (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

At some point, you may need to change [data types](#) for fields that already contain data. Perhaps you imported data, and Microsoft Access didn't set the data types as you intended. Or maybe the data type you set for a field is no longer appropriate.

Before converting from one data type to another, consider how the change will affect your entire database. Which queries, forms, and reports use the field that is being converted? You may need to change [expressions](#) that depend on the changed field.

The most common data type changes fall into the following four categories: converting to Text from other data types; converting from Text to Number, Currency, Date/Time, or Yes/No; converting between Currency and Number or between Text and Memo; and changing the **FieldSize** property setting for Number fields.

[Converting to Text from other data types](#)

While you should store numbers that will be used in calculations in a Number or Currency field, you may find that a field that was originally just numbers now requires the use of letters and other non-numeric characters. For example, suppose you have a Number field that stores code numbers. If you later find you need to include letters, hyphens, parentheses, or other non-numeric characters as part of the code, you'll need to change the field's data type to Text.

Microsoft Access converts number values to text by using a General Number format, and date values to text by using a General Date format. The converted values will not include any currency symbols or other special formatting characters you set for the field.

[Converting from Text to Number, Currency, Date/Time, or Yes/No](#)

If you have data stored in a Text field and want to change the field to another data type, Microsoft Access can convert values that are appropriate for the new data type. For example, if you have numbers stored in a Text field and you want to do mathematical calculations on your data, you must convert the field to the Number or Currency data type. As long as all the data stored in the field consists of only numbers, you can change the data type without losing data.

For Text to Number data type conversions, decimal points and thousands separators are interpreted appropriately. Currency symbols are interpreted according to the regional settings in Microsoft Windows Control Panel.

For Text to Date/Time conversions, most date and time formats are converted correctly. Date and time formats are interpreted according to the regional settings in Windows Control Panel.

For Text to Yes/No conversions, the words Yes, True, or On are converted to a Yes value, and No, False, or Off to a No value. To display the words "Yes" or "No" instead of checkboxes, you must click the **Lookup** tab in table [Design view](#) and change the **DisplayControl** property from **Check Box** to **Text Box**. (You can also convert from Number to Yes/No data type: zero or **Null** values convert to No and nonzero values to Yes.)

[Converting between Currency and Number or between Text and Memo](#)

You can always convert between the Currency and Number data types. You should always store monetary values using the Currency data type to ensure that calculations are correct. You should also use a Currency data type if you plan to perform many calculations on a field that contains numbers with one to four decimal places. Currency fields use a fixed-point calculation method that avoids rounding errors.

You can always convert a Text field to a Memo field. For example, if you want to store longer text values than you originally intended, you can change a field to the Memo data type. However, if a field doesn't have to store more than 255 characters, you should use the Text data type.

[Changing the field size of Number fields](#)

For fields with the Number data type, the **FieldSize** property setting determines the specific number type: **Byte**, **Integer**, **Long Integer**, **Single**, **Double**, **Decimal**, or **Replication ID** (GUID). If you convert a field to the Number type, consider whether you need to change the **FieldSize** setting for the values you store in the field. If you change the **FieldSize** setting from a larger size, such as **Double**, to a smaller size, such as **Integer**, you must make sure that the values stored in the field fit in the new field size. If you change to a **FieldSize** setting that doesn't allow the number of decimal places included in your current values, the numbers are rounded. For example, if you change a field from **Double** to **Long Integer**, decimal numbers are rounded to the nearest whole number. Also, when you attempt to save a table containing values that are too large to be stored in the new size, Microsoft Access warns you that they will be deleted and replaced with **Null** values if you proceed.

[Field data type conversion results](#)

The following is a list of the results of common data type conversions when the table contains data.

From All data types to AutoNumber

Prohibited by Microsoft Access.

From Text to Number, Currency, Date/Time, or Yes/No

Converts text to appropriate values. Make sure values fit in the new data type; inappropriate values are deleted.

From Memo to Text

Simple conversion. Data longer than the **FieldSize** setting is truncated.

From Number to Text

Converts values to text. Numbers adopt General Number format.

From Number to Currency

Converts numbers to currency. Make sure values fit in the new data type; inappropriate values are deleted.

From Date/Time to Text

Converts values to text. Dates or times adopt General Date format.

From Currency to Text

Converts values to text. Text doesn't include currency symbols, such as \$.

From Currency to Number

Simple conversion. Make sure values fit in the new data type; inappropriate values are deleted.

From AutoNumber to Text

Converts values to text. Values may be truncated depending on **FieldSize** setting.

From AutoNumber to Number

Simple conversion. Make sure values fit in the new data type; inappropriate values are deleted.

From Yes/No to Text

Converts values to text.

[Show All](#)

Set or change a field's data type or size (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

If your table contains data, make a backup copy of the table before you change [data types](#) or field sizes.

[Change a field's data type](#)

1. Open the table in [Design view](#).
2. Click the **Data Type** column of the field you want to change, click the arrow, and select the new data type.
3. Click **Save**  on the toolbar.

Caution If the data type conversion would result in lost values, Microsoft Access displays a message telling you that errors occurred during conversion before it actually saves the changes. Click **Cancel** to cancel the changes. Click **OK** to continue and save the changes anyway.

Note In large tables, changing a data type might take a long time. If you want to cancel the conversion process at any point while it is running, press CTRL+BREAK, and then click **OK**.

[Change the field size of a Text or Number field](#)

1. In table [Design view](#), in the upper portion of the window, click the field whose **FieldSize** property you want to set.
2. In the lower portion of the window on the **General** tab, click the **FieldSize** property box.
3. Do one or more of the following:

For a Text field, type the maximum number of characters to allow in the field (up to 255).

For a **Number** field, click the arrow and select the desired field size.

For more information on field sizes, click the **FieldSize** property box and press F1.

[Change the default field size of Text and Number fields](#)

1. On the **Tools** menu, click **Options**, and click the **Tables/Queries** tab.
2. Under **Default field sizes**, do one or more of the following:

In the **Text** box, type the maximum number of characters to allow in the field (up to 255).

In the **Number** box, click the arrow and select the desired field size.

[Show All](#)

About creating a field that looks up or lists values in tables (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

A Lookup field provides a list of values that you can choose from when you are entering data. This makes data entry easier and ensures the consistency of the data in that field. A Lookup field can get its list of values from a table or query, or from a fixed set of values that you have specified.

 [Lookup fields that display values from a table or query](#)

The most common Lookup list displays values looked up from a related table. For example, the SupplierID field in the Products table displays this Lookup list:



ID	Supplier
42	Leka Trading
43	Leka Trading
44	Lyngbysild
45	Ma Maison
46	Mayumi's

1 Lookup list

This list is created by looking up the SupplierID values in the Suppliers table and displaying the corresponding Supplier names. Picking a value from a Lookup list sets the [foreign key](#) value in the current record (SupplierID in the Products table) to the [primary key](#) value of the corresponding record in the related table (SupplierID in the Suppliers table). This creates an association to the related table to display (but not store) the Supplier names in the record. The foreign key (SupplierID) is stored but not displayed. For this reason, any updates made to the data in the Suppliers table will be reflected in both the list and records in the Products table. You must define a Lookup list field from the table that will contain the foreign key and display the Lookup list. In the above example, the Lookup list field would be defined from the Products table.

[Lookup fields that display a fixed set of values](#)

A value list looks the same as a Lookup list, but consists of a fixed set of values you type in when you create it. A value list should only be used for values that will not change very often and don't need to be stored in a table. For example, a list for a Salutation field containing Mr., Mrs., or Ms. would be a good candidate for a value list. Choosing a value from a value list will store that value in the record—it doesn't create an association to a related table. For this reason, if you change any of the original values in the value list later, they will not be reflected in records added before this change was made.

[Forms based on tables with Lookup fields](#)

Once you've created a Lookup list field, if you add the field to a form, Microsoft Access copies its definition into the form. The combo box or list box is created automatically for the form. However, if you change the definition of a Lookup or value list field in the table after adding it to a form, those changes will not be reflected in that form. To correct this, delete the field from the form and then add it again.

[Creating a field to look up data in the same table](#)

You can create in a table a Lookup field that displays values from the same table. For example, in the Employees table, you can display the name of each employee's supervisor in the ReportsTo field—a Lookup field that displays data from the FirstName and LastName fields by looking up the corresponding EmployeeID.

[Show All](#)

Create a field that looks up or lists values in tables (MDB)

You can add a new [Lookup](#) or value list field in either table [Design view](#) or table [Datasheet view](#). However, if the field you want to use as the [foreign key](#) for a Lookup field already exists, you must open that field's table in Design view to define the Lookup field. For example, if you have a Products table that has a SupplierID field already defined, and you want to change it to a Lookup field to display supplier names from your Suppliers table, you must open the Products table in Design view to change SupplierID to a Lookup field.



[In Design view, create a field that looks up data from another table](#)

1. In Design view, open the table where you want to add the Lookup field.
2. Do one of the following:

To insert a new field within the table, click in the row below where you want to add the field, and then click **Insert Rows**  on the toolbar, or to add a new field at the end of the table, click in the first blank row. Type a name for the field in the **Field Name** column.

If the field you want to use as the foreign key for the Lookup field already exists, click in that field's row. For example, if you have a Products table that has a SupplierID field already defined, and you want to change it to a Lookup field to display supplier names from your Suppliers table, click in the SupplierID field's row.

3. In the **Data Type** column, click the arrow and select **Lookup Wizard**.
4. In the first dialog box of the Lookup Wizard, select the option that indicates you want the Lookup field to look up the values in a table or query.
5. Click **Next** and follow the directions in the remaining **Lookup Wizard** dialog boxes.

When you click the **Finish** button, Microsoft Access creates a Lookup field whose properties are based on the choices you made in the wizard.

[In Datasheet view, create a field that looks up data from another table](#)

1. In Datasheet view, open the table where you want to add the Lookup field.
2. Click in the column to the right of where you want to insert the field, and then on the **Insert** menu, click **Lookup Column**.
3. In the first dialog box of the Lookup Wizard, select the option that indicates you want the Lookup field to look up the values in a table or query.
4. Click **Next** and follow the directions in the remaining **Lookup Wizard** dialog boxes.

Microsoft Access sets properties of the field based on the choices you made in the wizard. By default, the name of the new Lookup field is "Field#," where # is some number. If you want, you can change the name of the field.

[In Design view, create a value list field](#)

1. Open the table in [Design view](#).
2. Do one of the following:

To insert the field within the table, click in the row below where you want to add the field, and then click **Insert Rows**  on the toolbar.

To add the field to the end of the table, click in the first blank row.

3. In the **Field Name** column, type a unique name for the field.
4. In the **Data Type** column, click the arrow and select **Lookup Wizard**.
5. In the first **Lookup Wizard** dialog box, click the option that indicates you will type in the values that you want.
6. Click **Next** and follow the directions in the remaining **Lookup Wizard** dialog boxes.

Microsoft Access sets properties of the field based on the choices you made in the wizard.

[In Datasheet view, create a value list field](#)

1. Open the table in Datasheet view.
2. Click in the column to the right of where you want to insert the field, and then on the **Insert** menu, click **Lookup Column**.

3. In the first dialog box of the **Lookup Wizard**, select the option that indicates that you will type in the values that you want.
4. Click **Next** and follow the directions in the remaining **Lookup Wizard** dialog boxes.

Microsoft Access sets the properties of the field based on the choices you made in the wizard. By default, the name of the new Lookup field is "Field#," where # is some number. If you want, you can change the name of the field.

[Show All](#)

Move a column (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Important Changing the field order in table or query [Design view](#) can affect combo boxes or list boxes that are based on that table or query. If your table or query is the record source for a combo box or list box, consider moving the field in [Datasheet view](#).

[Move a column in Datasheet view](#)

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#).
2. Select the columns you want to move.

To select one column, click the [field selector](#) for that column.



Last Name:
Davolio
Fuller
Leverling

To select adjacent columns, click a column field selector and then, without releasing the mouse button, drag to extend the selection.

3. Click and hold down the mouse button in the field selectors again.
4. Drag the columns to a new position.

Note If the columns don't move, they may be frozen, and you may have to unfreeze them.

[Move a field in table Design view](#)

Changing field order in table Design view changes the order in which fields are stored in the table and also changes the column order in the table's [datasheet](#).

1. Open a table in Design view.
2. Select the fields you want to move.

To select one field, click that field's [row selector](#).

To select a group of fields, drag through the row selectors of those fields.



1 Row selectors

3. Click and hold down the mouse button in the row selector again. Microsoft Access displays a thin horizontal bar just above the last selected row.
4. Drag the horizontal bar to the row just below where you want to move the fields.

[Move a field in the design grid in a query or advanced filter](#)

1. Open a query in [Design view](#), or open a form or datasheet and show the [Advanced Filter/Sort window](#).
2. Do one of the following:

To select one column, click its [column selector](#).

To select more than one adjacent column, drag across their column selectors.

3. Click the selector of any one of the selected columns again, and then drag the field or fields to their new location.



1 A selected column

2 The pointer changes to a rectangle during the move

[Show All](#)

About renaming a field in a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To rename a field in a table, you can change the name of the field itself, or you can change the **Caption** property of the field to specify a different display name. The **Caption** property determines the field name that appears in [Datasheet view](#). It also specifies the default text for labels attached to [controls](#) in forms and reports. The name specified by the **Caption** property might be different from the name of the field itself.

This is useful in situations where you want the column heading to be longer or more descriptive than the actual field name. For example, you might want to name a field LName for use in [Microsoft Visual Basic for Applications](#) code and [expressions](#), but display "Last Name" as the field's column heading.

If you don't specify a caption for a field in a table, the name of the field is used.

[Show All](#)

Rename a field in a table or datasheet (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

If the **Caption** property has been set for a field, the text displayed in the [field selector](#) might be different from the actual field name. If you rename a field with a caption while in [Datasheet view](#), the caption will be deleted and replaced with the field name. To avoid deleting a caption, rename the field in [Design view](#).

 [Rename a field in a table in Design view](#)

Important If you have [expressions](#) or [Microsoft Visual Basic for Applications](#) code that contains references to a field whose name you are changing, you must update all of those references to the new name.

1. Open the table in Design view.
2. Double-click the field name you want to change.
3. Type the new field name.
4. Click **Save**  on the toolbar to save your changes.

 [Rename a field in a table in Datasheet view](#)

Important If you have [expressions](#) or [Microsoft Visual Basic for Applications](#) code that contains references to a field whose name you are changing, you must update all of those references to the new name.

1. Open the table in Datasheet view.
2. Double-click the field selector of the field that you want to change.
3. Type the new field name and press ENTER to save it.

 [Change the field \(column\) name that appears in a datasheet](#)

1. Open the table in Design view.

2. In the upper portion of the window, click the field for which you want to define a different column name.
3. In the lower portion of the window, click the **Caption** property box, and then type the text for the column name. The text can contain any combination of letters, numbers, spaces, and symbols up to 255 characters in length.

[Show All](#)

Set the display format for data in a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Set the data display format for a field in a table](#)

1. Open a table in [Design view](#).
2. In the upper portion of the window, click the field you want to format.
3. In the lower portion of the window, do one of the following:
 - Click the arrow next to the **Format** property box, and then click one of the predefined formats.
 - Type a custom format in the **Format** property box for any field data type other than [OLE Object](#). Click the **Format** property box and press F1 for more information about custom formats.

[Set the data display format for a field in a query](#)

1. Open a query in [Design view](#).
2. In the query design grid, place the insertion point in the column for the field you want to format. You can place the insertion point in any row for that field.
3. Click **Properties**  on the **Query Design** toolbar to open the property sheet for that field.
4. On the **General** tab, do one of the following:
 - Click the arrow next to the **Format** property box, and then click one of the predefined formats.
 - Type a custom format in the **Format** property box for any field data type other than [OLE Object](#). Click the **Format** property box and press F1 for more information about custom formats.

[Set the data display format for a control on a form, report, or data access page](#)

1. Open the form, report, or [data access page](#) in [Design view](#).
2. Click the [control](#) you want to change, and then click **Properties**  on the **Form Design, Report Design, or Page Design** toolbar to open its property sheet.
3. In the **Format** property box, do one of the following:
 - Click one of the predefined formatting settings.
 - Type a custom format. Click the **Format** property box and press F1 for more information about custom formats.

Note To ensure consistency among applications, Microsoft Access uses the regional settings that are specified in Microsoft Windows Control Panel for some predefined Number and Date/Time formats.

[Show All](#)

Set the default control type for a field (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can set a default [control](#) type for Text, Number, and Yes/No fields. For Text and Number fields, the default control type can be a text box, combo box, or list box (forms only). For Yes/No fields, the default control type can be a check box, text box, or combo box. Once you have set a field's default control type, Microsoft Access uses that type of control to display the field's data in the datasheet for a table. Also, it automatically creates that type of control whenever you add that field to a form or report.

1. Open the table that contains the field in [Design view](#).
2. Click in the **Data Type** column for a Text, Number, or Yes/No field.
3. Click the **Lookup** tab in the lower portion of the screen.
4. In the **DisplayControl** property box, click a control type in the list.

[Show All](#)

About setting the number of decimal places to display in a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In a [Microsoft Access database](#), you'll usually want to set the property for the field in table [Design view](#). This way, the setting is automatically applied to the field in query Design view and to [controls](#) on a form or report [bound](#) to that field. In certain cases, however, you may want to set the property in query Design view, form Design view, or report Design view. For example, you may want to display a different number of decimal places in a table datasheet than in a report. You could set the **DecimalPlaces** property for the field in query Design view and then base your report on the query. If you're using an [unbound control](#), you'll need to set the **DecimalPlaces** property in form Design view or report Design view.

The **DecimalPlaces** property setting affects only the number of decimal places that are displayed, not how many decimal places are stored. For example, a field whose **FieldSize** property is set to **Single** can store numbers with seven digits to the right of the decimal point. However, if the field's **Format** property is set to **Fixed** and the **DecimalPlaces** property is set to **3**, you see only three digits to the right of the decimal point when you view the data. For instance, the stored value of 15.5555 is displayed as 15.556.

[Show All](#)

About property sheets

In Microsoft Access, you use properties to determine the characteristics of tables, [queries](#), fields, forms, reports, and [data access pages](#). Every section and [control](#) on a form, report, or data access page has properties as well. Control properties determine the structure, appearance, and behavior of a control, as well as the characteristics of the text or data it contains.

You set properties by using the property sheet. For example, the following illustration shows the property sheet for a TotalSales control on a report. The **Format** property for this control is set to **Currency**.



- 1 The title bar shows the type and name of the selected object.
- 2 Each tab displays properties of a specific category.
- 3 The values displayed in the TotalSales control will be formatted as currency.

[Show All](#)

Set the number of decimal places to display in a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The **DecimalPlaces** property provides a default setting, **Auto**, and the option to specify from 0 to 15 decimal places. When set to **Auto**, fields with **Format** property settings of **Currency**, **Fixed**, **Standard**, **Percent** and **Scientific** display two decimal places. The **DecimalPlaces** property has no effect unless the field or control has a **Format** property setting.



[Set the number of decimal places to display for a field in table Design view](#)

1. Open a table in [Design view](#).
2. In the upper portion of the window, click the field you want to define decimal places for.
3. In the lower portion of the window, click the arrow next to the **DecimalPlaces** property box, and then click the desired number of decimal places.



[Set the number of decimal places to display for a field in query Design view](#)

1. Open a query in [Design view](#).
2. In the query design grid, place the insertion point in the column for the field you want to change. You can place the insertion point in any row for that field.
3. Click **Properties**  on the **Query Design** toolbar to open the property sheet for that field.
4. On the **General** tab, click the arrow next to the **DecimalPlaces** property box, and then click the desired number of decimal places.



[Set the number of decimal places to display for a control in form Design](#)

[view or report Design view](#)

1. Open a form or report in [Design view](#).
2. Make sure the [control](#) you want to change is selected, and then click **Properties**  on the **Form Design** or **Report Design** toolbar to open the property sheet for the control.
3. Click the **DecimalPlaces** property box, click the arrow, and then select the desired number of decimal places.

Note This procedure does not apply to data access pages.

[Show All](#)

Change the starting value of an AutoNumber field (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

For a new table that contains no records, you can change the starting value of an [AutoNumber](#) field whose **NewValues** property is set to **Increment** to a number other than 1. For a table that contains records, you can also use this procedure to change the next value assigned in an AutoNumber field to a new number.

1. If your original table contains property settings that prevent [Null](#) values in fields, you must temporarily change those properties. These settings include:
 - The **Required** field property set to **Yes**
 - The **Indexed** field property set to **Yes (No Duplicates)**
 - A field and/or record **ValidationRule** property that prevents **Null** values in fields
2. Create a temporary table with just one field: a Number field. Set its **FieldSize** property to **Long Integer** and give it the same name as the AutoNumber field in the table whose value you want to change.
3. In [Datasheet view](#), enter a value in the Number field of the temporary table that is one (1) less than the starting value you want for the AutoNumber field. For example, if you want the AutoNumber field to start at 100, enter 99 in the Number field.
4. Create and run an [append query](#) to append the temporary table to the table whose AutoNumber value you want to change.

[How?](#)

1. Create a query that contains the table whose records you want to append to another table.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the toolbar, and then click **Append**. The **Append** dialog box appears.
3. In the **Table Name** box, enter the name of the table you want to append records to.
4. Do one of the following:

If the table is in the currently open database, click **Current Database**.

If the table is not in the currently open database, click **Another Database** and type the path of the database where the table is stored or click **Browse** to locate the database. You can also specify a path to a Microsoft FoxPro, Paradox, or dBASE database, or a connection string to an [SQL database](#).

5. Click **OK**.
6. Drag from the [field list](#) to the query [design grid](#) the fields you want to append and any fields you want to use for setting criteria.

If all the fields in both tables have the same names, you can just drag the asterisk (*) to the query design grid. However, if you're working in a database [replica](#), you'll need to add all the fields instead.

7. If you have a field with an AutoNumber [data type](#), do one of the following:

[Add AutoNumber values automatically](#)

To have Microsoft Access add AutoNumber values automatically,

don't drag the AutoNumber field to the query design grid when you create the query.

With this method, Access appends records and automatically inserts AutoNumber values. The first record appended has a value that is one larger than the largest entry that was ever entered in the AutoNumber field (even if the record that contained the largest AutoNumber value has been deleted).

Use this method if the AutoNumber field in the table you're appending to is a [primary key](#), and the original table and the table you're appending to contain duplicate AutoNumber values.

[Keep the AutoNumber values from the original table](#)

To keep the AutoNumber values from the original table, drag its AutoNumber field to the query design grid when you create the query.

8. If the fields you've selected have the same name in both tables, Microsoft Access automatically fills the matching name in the **Append To** row. If the fields in the two tables don't have the same name, in the **Append To** row, enter the names of the fields in the table you're appending to.
9. In the **Criteria** cell for the fields that you have dragged to the grid, type the criteria on which additions will be made.
10. To preview the records that the query will append, click **View**  on the toolbar. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
11. Click **Run**  on the toolbar to add the records.
5. Delete the temporary table.
6. Delete the record added by the append query.
7. If you had to disable property settings in step 1, return them to their original settings.

When you enter a record in the remaining table, Microsoft Access uses an AutoNumber field value one (1) greater than the value you entered in the temporary table.

[Show All](#)

Set a property in Design view

1. Open a table, query, form, report, or [data access page](#) in [Design view](#).
2. Open the property sheet by doing one of the following:
 - On a form or report, double-click the [form selector](#) or the [report selector](#).



1 Form or report selector

- On a data access page, click **Select Page** on the **Edit** menu.
- On a table or query, click **Properties**  on the toolbar.
- For a section on a form or report, double-click the [section selector](#).



1 Section selector

- For a section on a data access page, double-click the section bar.



1 Section bar

- To set properties of a query field, click the cell in the **Field** row, and then click **Properties**  on the toolbar.
 - To set properties of a query field list, click anywhere in the list, and then click **Properties**  on the toolbar.
 - To set properties for the body of a data access page, click above the first section, and then click **Properties**  on the toolbar.
 - To set properties for a [control](#), click the control, and then click **Properties**  on the toolbar.
3. In the property sheet, click the property you want to set, and then do one of

the following:

- If an arrow appears in the property box, click it and then select a value from the list.
- Type a setting or [expression](#) in the property box.
- If a **Build** button  appears next to the property box, click it to display a [builder](#) or list of builders.

[Tips](#)

- To get Help on any property in the property sheet, click the property and then press F1.
- If you need more space to enter or edit a property setting, press SHIFT+F2 to open the **Zoom** box.

Note If the **AllowDesignChanges** property of a form is set to **All Views**, you can also set form properties in [Form view](#) and [Datasheet view](#).

[Show All](#)

About defining a default value that is automatically entered in a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can use the **DefaultValue** property to specify a value that will be automatically entered when you add a new record. For example, if most of your suppliers are in New York, you could set the City field of the Suppliers table to have a default value of "New York." You can accept the default value or type a new value over it.

In most cases, you'll want to add the default value to the field in table [Design view](#), since that default value will be applied to [controls](#) based on that field. If the control is [unbound](#), however, or if it's based on data from a linked (external) table, you'll need to set the default value for the control in the form or data access page.

[Show All](#)

Set a default value for a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In a [Microsoft Access database](#), you can define a default value in table [Design view](#), and in a Yes/No field, you can set a default value for international use. In a [Microsoft Access file](#), you can define a default value for a [control](#) on a form or data access page in Design view.

[Set a default value for a field in a table](#)

1. Open a table in Design view.
2. In the upper portion of the window, click the field you want to define a default value for.
3. In the lower portion of the window, click the **DefaultValue** property box, and then type a default value such as **New York**, or an [expression](#) such as **Date()**.

Note Setting the **DefaultValue** property for a field has no effect on existing data.

[Set the default value of a Yes/No field so that it is displayed in the local language](#)

1. Open a table in Design view.
2. In the upper portion of the window, click the field you want to define a default value for.
3. In the lower portion of the window, click the **Lookup** tab and set the **DisplayControl** property for the field to **Text Box** or **Combo Box**.
4. On the **General** tab, click the **DefaultValue** property box and type =**Yes** or =**No**.

[Set a default value for a control on a form or data access page](#)

1. Open a form or data access page in [Design view](#).
2. Make sure the [control](#) is selected, and then click **Properties**  on the toolbar to open the control's property sheet.
3. Click the **Data** tab, and in the **DefaultValue** property box, type the default value (such as **New York**) or an [expression](#) (such as =**Date()**). If the string expression that you type includes commas or other punctuation, be sure to enclose the value in quotation marks. For example, to use "New York, NY" as the default value, type "**New York, NY**". In a form in a [Microsoft Access database](#), if you need help creating an expression, click the **Build** button  to use the [Expression Builder](#).

Note If you set the **DefaultValue** property for a field in table Design view and also set the **DefaultValue** property for a control bound to that field, the default value set for the control overrides the default value set for the field.

[Show All](#)

Examples of default values in fields (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Expression	Default field value
1	1
"MT"	MT
"New York, N.Y."	New York, N.Y. (note that you must enclose the value in quotes if it includes punctuation)
" "	Zero-length string
Date()	Today's date
=Yes	Yes displayed in the local language of the computer.

[Show All](#)

Distinguish between Null values and zero-length strings (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When you view data in a field that contains both **Null** values and [zero-length strings](#), the fields look the same— they contain no values. However, you can use an [expression](#) to distinguish **Null** values from zero-length strings.

1. Open a query, form, report, or data access page in [Design view](#).
2. In a field in the [query design grid](#), or in the control source of an [unbound control](#), type the following expression. Substitute for *fieldname* the name of the field that contains both **Null** values and zero-length strings.

=IIf(IsNull([*fieldname*]),"Unknown",Format([*fieldname*],"@;\ZLS"))

The expression returns "Unknown" if the field contains **Null** and "ZLS" if the field contains a zero-length string; otherwise, the expression returns the value in that field.

[Show All](#)

About blank values in fields

Microsoft Access allows you to distinguish between two kinds of blank values: **Null** values and [zero-length strings](#). In some situations, a blank value indicates that the information might exist but is not currently known. In other situations, a blank value indicates that the field does not apply to a particular record.

For example, if a Customers table contains a Fax Number field, you can leave the field blank if you're unsure of the customer's fax number, or whether the customer even has a fax. In this case, leaving the field blank enters a **Null** value, which means you don't know what the value is. If you later determine that the customer doesn't have a fax, you can enter a zero-length string in the field to indicate that you know there is no value.

[Show All](#)

Make a field accept zero-length strings (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Only Text, Memo, and [Hyperlink](#) fields can accept [zero-length strings](#).

1. Open the table that contains the field in [Design view](#).
2. Click the field in which you want to enter a zero-length string.
3. Set the **AllowZeroLength** field property to **Yes**.

[Show All](#)

Set properties that control blank values (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can specify the types of blank values that are allowed in a field by setting different combinations of the field's **Required** and **AllowZeroLength** properties. Microsoft Access allows you to distinguish between two kinds of blank values: **Null** values and [zero-length strings](#). The **Required** property determines whether users can leave a field blank, resulting in a **Null** value. The **AllowZeroLength** property determines whether Text, Memo, or [Hyperlink](#) fields can contain a zero-length string.

1. In table [Design view](#), in the upper portion of the window, click the field for which you want to allow or prevent blank values.
2. In the lower portion of the window, do one of the following:

To allow blank values when you don't need to distinguish blank values that indicate unknown data from blank values that indicate you know there's no value, set both the **Required** and **AllowZeroLength** properties to **No**.

To prevent users from leaving a field blank, set the **Required** property to **Yes** and the **AllowZeroLength** property to **No**.

To allow both types of blank values, so that you can distinguish blank values that indicate unknown data from blank values that indicate you know there's no data, set the **Required** property to **No** and the **AllowZeroLength** property to **Yes**.

To allow blank values in a field only when you know that there is no data, set both the **Required** property and the **AllowZeroLength** property to **Yes**. In this case, the only way to leave a field blank is to type double quotation marks with no space between them, or press the SPACEBAR to enter a zero-length string.

[Show All](#)

About restricting or validating data (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Microsoft Access provides a variety of ways to control how users enter data into your database. For example, you can limit the data that a user can enter into a [field](#) by defining a [validation rule](#) for that field. If the data that a user enters into the field breaks the rule, Access will display a message telling the user what kind of entries are allowed. Another method of controlling data entry is to create an [input mask](#) to restrict the kind of values that can be entered in positions across the field. You can perform these simple forms of validation and restriction by setting [properties](#) for fields in [tables](#), or by setting properties for [controls](#) on [forms](#).

In most cases, it is preferable to define data validation and restriction by setting a field's properties in table [Design view](#). That way, whenever you use that field in a form, the field's validation rule and other properties will apply to data entry performed using the form.

However, if the data entered into a control on a form isn't [bound](#) to a field in a table, and you need to restrict or validate data entry, you must define those properties in the form. Additionally, there are situations where you must use [macros](#) or [Microsoft Visual Basic for Applications](#) code in conjunction with a form to perform more complex validation. For example, you might want to be able to override your validation rule or compare values from different tables.



[Using data types and field properties to restrict or validate data in tables](#)

Microsoft Access provides a variety of ways to control how data is entered in fields in your table.

The first thing you should consider is whether a field has an appropriate [data type](#). For example, Date/Time fields only allow entry of valid date and time formats. If you are currently using a Text field for such information, it may be

appropriate to change the field to a Date/Time field.

You can set field properties that provide further control over the values that can be entered into the field:

- For Number fields, you can choose from a variety of field sizes to control the kind and range of values that can be entered.
- For Text fields, you can set the maximum number of characters that can be entered.
- For all but [AutoNumber](#) fields (which generate their own data), you can require that data be entered in the field.
- For Text, Date/Time, and Number fields, you can define an input mask to provide blanks to fill in and can control what values can be entered in those blanks.
- For all but [OLE Object](#) fields, you can prevent duplicate values from being entered into a field or combination of fields.

[Validation rules](#)

Validation rules allow you to define a rule to limit what will be accepted. They are enforced whenever you add or edit data, whether it is through table [Datasheet view](#), a form bound to the table, an [append query](#), an [update query](#), Visual Basic for Applications code, or by importing data from another table. You can define two kinds of validation rules: field validation rules and record validation rules.

- A field validation rule is used to check the value entered into a field as the user leaves the field. For example, you could define ">=10 And <=100" as the validation rule for a Number field to allow only values from 10 to 100 to be entered.
- A record validation rule controls when an entire record can be saved. Unlike field validation rules, record validation rules can refer to other fields in the same table. This makes them useful when you want to compare values in different fields. For example, you could define "[RequiredDate]<=[OrderDate]+30" as the validation rule for an Orders table. This rule would make sure that the date entered into the RequiredDate field is within 30 days of the date in the OrderDate field.

When a field or record validation rule is broken, Access displays a message that informs the user how to properly enter data.

[Using a macro or event procedure for data validation](#)

In most cases, you can verify that data is entered correctly into a control on a form by setting the **ValidationRule** property for the control, or by setting record or field validation rules in the underlying table to which the control is bound. Macros and [event procedures](#) provide additional power and flexibility for more complex validation.

Use a macro or an event procedure for data validation if:

- Your validation rule involves conditions for more than one value on the form. For example, you might want to verify that any two of three fields are filled in before saving a record.
- You want to display different error messages for different types of errors in the field. For example, if the value entered is greater than the acceptable range, you can display one message, and if it's less than the acceptable range, you can display a different message.
- You want the user to be able to override your validation rule. In this case, you can display a warning message asking the user to confirm the data.
- The validation involves references to controls on other forms or contains a function.
- You have a validation rule that you can use for more than one form, and you want the convenience of defining the rule once and then referring to it in each form.

[Show All](#)

Examples of field validation rules (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

ValidationRule setting	ValidationText setting
<>0	Please enter a nonzero value.
0 or >100	Value must be either 0 or over 100.
<#1/1/2000#	Enter a date before 2000.
>=#1/1/2000# and <#1/1/2001#	Date must be in 2000.
StrComp(UCase([LastName]), [LastName],0) = 0	Data in the LastName field must be uppercase.

You can use also wildcard characters in a field validation rule. Which wildcard characters you use depends on the [ANSI SQL query mode](#) of the [Microsoft Access database](#).

The following example works in an Access database that uses Microsoft Jet SQL syntax.

ValidationRule setting	ValidationText setting
Like "K???"	Value must be four characters beginning with the letter K.

The following example works in an Access database that uses Microsoft SQL Server-compatible syntax (ANSI-92).

ValidationRule setting	ValidationText setting
Alike "K___"	Value must be four characters beginning with the letter K.

[Show All](#)

Validate or restrict data entry in tables (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

A field validation rule is used to check the value entered into a field as the user leaves the field. A record validation rule controls when an entire record can be saved. Unlike field validation rules, record validation rules can refer to other fields.

You can also determine whether data that already existed before you specified a validation rule or changed the **Required** or **AllowZeroLength** property violates the current settings.

1. Open a table in [Design view](#).
2. Do one of the following:

 [Define a validation rule to control what values can be entered into a field](#)

1. In the upper portion of the window, click the field you want to define a validation rule for.
2. In the lower portion of the window, click the **ValidationRule** property box, and then type the validation rule, or click the **Build** button  to create the validation rule using the [Expression Builder](#).

For example, you could define the validation expression ">9" for a Quantity field to prevent a user from placing an order for fewer than 10 units.

3. In the **ValidationText** property box, type the message that you want displayed when the rule is broken.

For example, for the validation expression ">9", you could enter "You must order 10 or more units."

4. If you set a validation rule in a field that contains data, Microsoft Access will ask if you want to apply the new rule to existing data when you save the table. If you click **Yes**, Access will warn you when existing data violates the validation rule.

[Define a validation rule to control when a record can be saved](#)

1. Click **Properties**  on the toolbar to display the table's property sheet.
2. In the **ValidationRule** property box, type the validation rule. Or click the **Build** button  to create the validation rule using the [Expression Builder](#).

For example, you could define the validation expression "[RequiredDate]<=[OrderDate]+30" to make sure that the date entered into the RequiredDate field is within 30 days of the date in the OrderDate field.

3. In the **ValidationText** property box, type the message that you want Microsoft Access to display when the rule is broken.

For example, for the validation expression "[RequiredDate]<=[OrderDate]+30", you could enter "The required date must be within 30 days of the order date."

4. If you set a validation rule in a table that contains data, Microsoft Access will ask if you want to apply the new rule to existing data when you save the table. If you click **Yes**, Microsoft Access will warn you when existing data violates the validation rule.

[Test data that already existed before you specified a validation rule](#)

- o Right click the title bar of the table, and click **Test Validation Rules**.

Microsoft Access will warn you if existing data violates a validation rule or the **Required** or **AllowZeroLength** settings.

[Show All](#)

Require users to enter data in a field (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. In table [Design view](#), in the upper portion of the window, click the field that you want to require data entry for.
2. In the lower portion of the window, click the **Required** property box, and then click **Yes**.

When you save changes to your table, Microsoft Access gives you the option to check to see whether the field has a value in all existing records. However, you can require that a value be entered in this field in all new records even if there are existing records with **Null** (blank) values in the field that aren't going to be changed.

Note You can set a field's **Required** property in table Design view only. However, Microsoft Access enforces the value of the field's **Required** property throughout the database. So if you set this property to **Yes**, the field must have a value whenever you enter data into the field: in the table, in forms and datasheets based on the table, when a [macro](#) or [Microsoft Visual Basic for Applications](#) code sets the value of the field, or when data is imported into the table.

[Show All](#)

About creating input masks to control how data is entered in a field or control (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can use the **InputMask** property to create an input mask (sometimes called a "field template") that uses literal display characters to control how data is entered in a field or [control](#). For example, the following input mask requires that all phone number entries contain exactly enough digits to make up a U.S. area code and telephone number, and that only numbers can be entered in each blank. You just fill in the blanks.

An input mask is used in fields (in tables and queries) and in text boxes and combo boxes (in forms) to format data and provide some control over what values can be entered. The **InputMask** property setting consists of literal characters (such as spaces, dots, dashes, and parentheses) along with special characters that determine the kind of value that can be entered into the blank in that position. Input masks are primarily used in Text and Date/Time fields, but can also be used in Number or Currency fields.

Phone
(206)555-1234
(206)5- - - -

If a field has an input mask, you enter data into the blanks.

You can set the **InputMask** property in [Design view](#) of a table, query, or form. In most cases, you'll want to set the property for the field in table Design view. This way, the input mask is automatically applied to the field in query Design view and to controls on a form or report [bound](#) to that field. In certain cases, however, you may want to set the property in query Design view or form Design view. For example, you might want to omit the input mask in a table but include it in a text box. Also, if you're using an [unbound control](#), you'll need to set the **InputMask** property in form Design view.

[Choosing between using an input mask and a display format](#)

Microsoft Access provides two field properties that produce similar results: the **Format** property and the **InputMask** property.

Use the **Format** property to display data in a consistent format. For example, if you set the **Format** property for a Date/Time field to **Medium Date** format, all dates entered will display in this form: 12-Jan-1999. If a user of your database enters a date as 01/12/99 (or any other valid date format), Microsoft Access will convert the display to the **Medium Date** format when the record is saved.

The **Format** property affects only how a value is displayed, not how it is stored in the table. Also, a [display format](#) isn't applied until the data entered is saved — nothing is displayed in the field to suggest or control the format in which data is entered. If you want data to display exactly as entered, don't set the **Format** property. If you need to control how data is entered, use an [input mask](#) in addition to, or instead of, a data display format. An input mask ensures that the data will fit in the format you define, and you can specify the kind of values that can be entered.

If you define both a display format and an input mask for a field, Microsoft Access uses the input mask when you are adding or editing data, and the **Format** setting determines how the data is displayed when the record is saved. When using both **Format** and **InputMask** properties, be careful that their results don't conflict.

[Show All](#)

Input mask syntax and examples

[Valid input mask characters](#)

Microsoft Access interprets characters in the **InputMask** property definition as shown in the following table. To define a literal character, enter any character other than those shown in the table, including spaces and symbols. To define one of the following characters as a literal character, precede that character with a backslash (\).

Character	Description
0	Digit (0 through 9, entry required; plus [+] and minus [-] signs not allowed).
9	Digit or space (entry not required; plus and minus signs not allowed).
#	Digit or space (entry not required; blank positions converted to spaces, plus and minus signs allowed).
L	Letter (A through Z, entry required).
?	Letter (A through Z, entry optional).
A	Letter or digit (entry required).
a	Letter or digit (entry optional).
&	Any character or a space (entry required).
C	Any character or a space (entry optional).
., : ; - /	Decimal placeholder and thousands, date, and time separators. (The actual character used depends on the regional settings specified in Microsoft Windows Control Panel.)
<	Causes all characters that follow to be converted to lowercase.
>	Causes all characters that follow to be converted to uppercase.
!	Causes the input mask to display from right to left, rather than from left to right. Characters typed into the mask always fill it from left to right. You can include the exclamation point anywhere in the input mask.
\	Causes the character that follows to be displayed as a literal character. Used to display any of the characters listed in this table as

literal characters (for example, \A is displayed as just A).

Setting the **InputMask** property to the word **Password** creates a Password password entry text box. Any character typed in the text box is stored as the character but is displayed as an asterisk (*).

[Input mask examples](#)

The following table shows some useful [input mask](#) definitions and examples of values you can enter into them.

Input mask definition	Examples of values
(000) 000-0000	(206) 555-0248 (206) 555-0248
(999) 999-9999!	() 555-0248
(000) AAA-AAAA	(206) 555-TELE -20
#999	2000 GREENGR339M3
>L????L?000L0	MAY R 452B7
>L0L 0L0	T2F 8M4 98115-
00000-9999	98115-3007 Maria
>L<????????????????	Pierre
ISBN 0-&&&&&&&&-0	ISBN 1-55615-507-7 ISBN 0-13-964262-5

>LL00000-0000

DB51392-0493

[Show All](#)

Create an input mask (MDB)

In a [Microsoft Access database](#), you can create an [input mask](#) in [Design view](#) of a table, query, or form.

[Create an input mask for a field in table Design view](#)

1. Open a table in Design view.
2. In the upper portion of the window, click the field you want to define an input mask for.
3. In the lower portion of the window, do one of the following:

Click the **Build** button  next to the **InputMask** property box to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.

Type the input mask definition. For more information on defining the input mask, click the **InputMask** property box and press F1. You must type the input mask definition manually for Number and Currency fields.

[Create an input mask for a field in query Design view](#)

1. Open a query in Design view.
2. In the query [design grid](#), place the insertion point in the column for the field you want to change. You can place the insertion point in any row for that field.
3. Click **Properties**  on the toolbar to open the property sheet for that field.
4. On the **General** tab, do one of the following:

Click the **Build** button  next to the **InputMask** property to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.

Type the input mask definition. For more information on defining the input mask, click the **InputMask** property box and press F1. You must type the input mask definition manually for Number and Currency fields.

[Create an input mask for a control on a form](#)

1. Open a form in [Design view](#).
2. Select a text box or a combo box, and then click **Properties**  on the toolbar to open the control's property sheet.
3. In the **Input Mask** property box, do one of the following:

Type the [input mask](#) definition.

In a [Microsoft Access database \(.mdb\)](#), click the **Build** button in the **Input Mask** property box to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.

[Show All](#)

About primary keys (MDB)

The power of a relational database system such as Microsoft Access comes from its ability to quickly find and bring together information stored in separate tables using [queries](#), [forms](#), and [reports](#). In order to do this, each table should include a field or set of fields that uniquely identifies each record stored in the table. This information is called the primary key of the table. Once you designate a primary key for a table, Access will prevent any duplicate or **Null** values from being entered in the primary key fields.

There are three kinds of primary keys that can be defined in Microsoft Access:

[AutoNumber primary keys](#)

An [AutoNumber](#) field can be set to automatically enter a sequential number as each record is added to the table. Designating such a field as the primary key for a table is the simplest way to create a primary key. If you don't set a primary key before saving a newly created table, Microsoft Access will ask if you want it to create a primary key for you. If you answer Yes, Microsoft Access will create an AutoNumber primary key.

[AutoNumber primary keys in a replicated database](#)

There are additional considerations if your table will be used with [database replication](#).

- If fewer than 100 records are routinely added between synchronizing [replicas](#), use a **Long Integer** setting for the **FieldSize** property to take up less disk space.

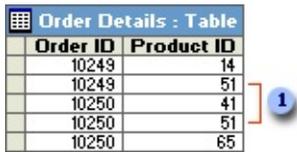
If more than 100 records are routinely added between synchronizing replicas, you should use **Replication ID** for the **FieldSize** property setting to prevent records from being assigned the same primary key value in each replica. Note, however, that an AutoNumber field with a Replication ID field size produces a 128-bit value that will require more disk space.

[Single-field primary keys](#)

If you have a field that contains unique values such as ID numbers or part numbers, you can designate that field as the primary key. You can specify a primary key for a field that already contains data as long as that field does not contain duplicate values or **Null** values.

[Multiple-field primary keys](#)

In situations where you can't guarantee the uniqueness of any single field, you may be able to designate two or more fields as the primary key. The most common situation where this arises is in the table used to relate two other tables in a [many-to-many relationship](#). For example, an Order Details table can relate the Orders and Products tables. Its primary key consists of two fields: OrderID and ProductID. The Order Details table can list many products and many orders, but each product can only be listed once per order, so combining the OrderID and ProductID fields produces an appropriate primary key.



Order ID	Product ID
10249	14
10249	51
10250	41
10250	51
10250	65

- 1 Each product can be listed only once per order.

Another example would be an inventory database that uses a field part number of two or more fields (part and subpart).

If you are in doubt about whether you can select an appropriate combination of fields for a multiple-field primary key, you should probably add an [AutoNumber](#) field and designate it as the primary key instead. For example, combining FirstName and LastName fields to produce a primary key is not a good choice, since you may eventually encounter duplication in the combination of these two fields.

In a multiple-field primary key, field order may be important to you. The fields in a multiple-field primary key are sorted according to their order in table [Design view](#). You can change the order of the primary key fields in the [Indexes window](#).

[Show All](#)

Set or change the primary key (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a table in [Design view](#).
2. Select the field or fields you want to define as the [primary key](#).

To select one field, click the [row selector](#) for the desired field.

To select multiple fields, hold down the CTRL key and then click the row selector for each field.

3. Click **Primary Key**  on the toolbar.

Note If you want the order of the fields in a multiple-field primary key to be different from the order of those fields in the table, click **Indexes**  on the toolbar to display the [Indexes window](#), and then reorder the field names for the [index](#) named PrimaryKey.

[Show All](#)

Prevent entry of duplicate values (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a table in [Design view](#).
2. Do one of the following:

[Prevent duplicate values from being entered into a field](#)

1. In the upper portion of the window, click the field that you want to prohibit duplicate values for.
2. In the lower portion of the window, click in the **Indexed** property box, and then click **Yes (No Duplicates)**.

[Prevent duplicate values from being entered into a combination of fields](#)

1. Create a multiple-field [index](#) using the fields you want to prohibit duplicate values for. Leave the [Indexes window](#) open when you have finished defining the index.

[How?](#)

1. Open the table in [Design view](#).
2. Click **Indexes**  on the toolbar.
3. In the first blank row in the **Index Name** column, type a name for the [index](#). You can name the index after one of the index fields, or use another name.
4. In the **Field Name** column, click the arrow and select the first field for the index.
5. In the next row in the **Field Name** column, select the second field for the index. (Leave the **Index Name** column blank in that row.) Repeat this step until you have selected all the fields you want to

include in this index.

Note The default [sort order](#) is **Ascending**. Select **Descending** in the **Sort Order** column of the [Indexes window](#) to sort the corresponding field's data in descending order.

2. In the upper portion of the Indexes window, click the new index name.
3. In the lower portion of the Indexes window, click the **Unique** property box, and then click **Yes**.

[Show All](#)

Remove the primary key (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. If the [primary key](#) is used in a [relationship](#), you must delete the relationship before you can remove the primary key.

 [How?](#)

1. Close any tables you have open. You can't delete [relationships](#) between open tables.
 2. Press F11 to switch to the [Database window](#).
 3. Click **Relationships**  on the toolbar.
 4. If the tables whose relationship you want to delete aren't displayed, click **Show Table**  on the toolbar and double-click each table you want to add. Then click **Close**.
 5. Click the relationship line for the relationship you want to delete (the line will turn bold when it's selected), and then press the DELETE key.
2. Open the table in [Design view](#).
 3. Click the [row selector](#) for the current primary key and then click **Primary Key**  on the toolbar.

Note This procedure doesn't delete the field or fields that are designated as the primary key; it simply removes the primary key features from the table.

[Show All](#)

View or edit indexes (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open the table in [Design view](#).
2. Click **Indexes**  on the toolbar.
3. Change [indexes](#) or index properties as desired. For more information about a property, click the property box and press F1.

[Show All](#)

About indexing fields and records in an Access database (MDB)

An index helps Microsoft Access find and sort records faster. Access uses indexes in a table as you use an index in a book: to find data, it looks up the location of the data in the index. You can create indexes based on a single field or on multiple fields. Multiple-field indexes enable you to distinguish between records in which the first field may have the same value.

Deciding which fields to index

You'll probably want to index fields you search frequently, fields you sort, or fields that you [join](#) to fields in other tables in [queries](#). However, indexes can slow down some [action queries](#) such as [append queries](#), when the indexes for many fields need to be updated while performing these operations.

The [primary key](#) of a table is automatically indexed, and you can't index a field whose [data type](#) is [OLE Object](#). For other fields, you should consider indexing a field if all the following apply:

- The field's data type is Text, Number, Currency, or Date/Time.
- You anticipate searching for values stored in the field.
- You anticipate sorting values in the field.
- You anticipate storing many different values in the field. If many of the values in the field are the same, the index may not significantly speed up queries.

Multiple-field indexes

If you think you'll often search or sort by two or more fields at a time, you can create an index for that combination of fields. For example, if you often set criteria for LastName and FirstName fields in the same query, it makes sense to create a multiple-field index on both fields.

When you sort a table by a multiple-field index, Microsoft Access sorts first by the first field defined for the index. If there are records with duplicate values in the first field, Microsoft Access sorts next by the second field defined for the index, and so on.

You can include up to 10 fields in a multiple-field index.

[Show All](#)

Create an index to find and sort records faster (MDB)

You can create [indexes](#) based on a single field or on multiple fields. Multiple-field indexes enable you to distinguish between records in which the first field may have the same value.

[Create a single-field index](#)

1. Open a table in [Design view](#).
2. In the upper portion of the window, click the field that you want to create an [index](#) for.
3. In the lower portion of the window, click in the **Indexed** property box, and then click **Yes (Duplicates OK)** or **Yes (No Duplicates)**.

[Create a multiple-field index](#)

1. Open the table in [Design view](#).
2. Click **Indexes**  on the toolbar.
3. In the first blank row in the **Index Name** column, type a name for the [index](#). You can name the index after one of the index fields, or use another name.
4. In the **Field Name** column, click the arrow and select the first field for the index.
5. In the next row in the **Field Name** column, select the second field for the index. (Leave the **Index Name** column blank in that row.) Repeat this step until you have selected all the fields you want to include in this index.

Note The default [sort order](#) is **Ascending**. Select **Descending** in the **Sort Order** column of the [Indexes window](#) to sort the corresponding field's data in descending order.

[Show All](#)

Delete an index (MDB)

1. Open the table in [Design view](#).
2. Click **Indexes**  on the toolbar.
3. In the [Indexes window](#), select the row or rows containing the [index](#) you want to delete and press the DELETE key.

Note This removes only the index, not the field itself.

[Show All](#)

View existing relationships (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Press F11 to switch to the [Database window](#).
2. Click **Relationships**  on the toolbar.
3. Do one of the following:

View all the relationships defined in the database

- Click **Show All Relationships**  on the toolbar.

View the relationships defined for a particular table

- Click the table, and then click **Show Direct Relationships**  on the toolbar.

View only the relationships defined for one table without viewing the relationships defined for other tables

1. Click **Clear Layout**  on the toolbar to remove all tables from the [Relationships window](#) (this won't delete the tables or [relationships](#); it just removes them from the Relationships window).
2. To add the table back, click **Show Table**  on the toolbar, double-click the table, and then click **Close**.
3. Click **Show Direct Relationships**  on the toolbar.

[Show All](#)

Edit an existing relationship (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Close any tables you have open. You can't modify [relationships](#) between open tables.
2. Press F11 to switch to the [Database window](#).
3. Click **Relationships**  on the toolbar.
4. If the tables whose relationship you want to edit aren't displayed, click **Show Table**  on the toolbar and double-click each table you want to add.
5. Double-click the relationship line for the relationship you want to edit.
6. Set the relationship options.

[Show All](#)

Remove a table from the Relationships window (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

- Click the table you want to remove, and then press the DELETE key.

This action affects only the display of the [Relationships window](#). The table and [relationships](#) remain in the database.

[Show All](#)

About relationships in an Access database (MDB)

After you've set up different tables for each subject in your [Microsoft Access database](#), you need a way of telling Microsoft Access how to bring that information back together again. The first step in this process is to define [relationships](#) between your tables. After you've done that, you can create queries, forms, and reports to display information from several tables at once. For example, this form includes information from four tables:

Product:	Unit Price:	Quantity:
Godgrandsblisst	\$24.00	14
Scottish Longbread	\$12.50	20
Pâté chinois	\$36.00	10
*		

- 1 The Customers table
- 2 The Orders table
- 3 The Products table
- 4 The Order Details table

[How relationships work](#)

In the previous example, the fields in four tables must be coordinated so that they show information about the same order. This coordination is accomplished with relationships between tables. A relationship works by matching data in key fields— usually a field with the same name in both tables. In most cases, these matching fields are the [primary key](#) from one table, which provides a unique identifier for each record, and a [foreign key](#) in the other table. For example, employees can be associated with orders they're responsible for by creating a

relationship between the EmployeeID fields.

Employees: Table		
Employee ID	Last Name	First Name
1	Davolio	Nancy

Orders: Table		
Order ID	Customer ID	Employee ID
10022	LAMAI	1

- 1 EmployeeID appears in both tables— -as a primary key ...
- 2 ... and as a foreign key.

[A one-to-many relationship](#)

A one-to-many relationship is the most common type of relationship. In a one-to-many relationship, a record in Table A can have many matching records in Table B, but a record in Table B has only one matching record in Table A.

Suppliers: Table	
Supplier ID	Company Name
1	Exotic Liquids
2	New Orleans Cajun Delights
3	Grandma Kelly's Homestead
4	Tokyo Traders

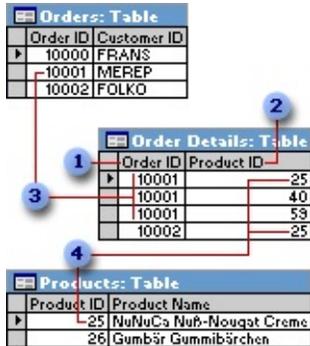
Products: Table		
Product ID	Product Name	Supplier ID
1	Chai	1
2	Chang	1
3	Aniseed Syrup	1
4	Chef Anton's Cajun Seasoning	2
5	Chef Anton's Gumbo Mix	2

- 1 One supplier ...
- 2 ... can supply more than one product ...
- 3 ... but each product has only one supplier.

[A many-to-many relationship](#)

In a many-to-many relationship, a record in Table A can have many matching records in Table B, and a record in Table B can have many matching records in Table A. This type of relationship is only possible by defining a third table (called a junction table) whose [primary key](#) consists of two fields— the [foreign keys](#) from both Tables A and B. A many-to-many relationship is really two one-to-many relationships with a third table. For example, the Orders table and the

Products table have a many-to-many relationship that's defined by creating two one-to-many relationships to the Order Details table. One order can have many products, and each product can appear on many orders.



- 1 Primary key from the Orders table
- 2 Primary key from the Products table
- 3 One order can have many products ...
- 4 ... and each product can appear on many orders.

A one-to-one relationship

In a one-to-one relationship, each record in Table A can have only one matching record in Table B, and each record in Table B can have only one matching record in Table A. This type of relationship is not common, because most information related in this way would be in one table. You might use a one-to-one relationship to divide a table with many fields, to isolate part of a table for security reasons, or to store information that applies only to a subset of the main table. For example, you might want to create a table to track employees participating in a fundraising soccer game. Each soccer player in the Soccer Players table has one matching record in the Employees table.



- 1 Each soccer player has one matching record in the Employees table.
- 2 This set of values is a subset of the EmployeeID field and the Employees table.

[About defining relationships](#)

The kind of relationship that Microsoft Access creates depends on how the related fields are defined:

- A one-to-many relationship is created if only one of the related fields is a [primary key](#) or has a [unique index](#).
- A one-to-one relationship is created if both of the related fields are primary keys or have unique indexes.
- A many-to-many relationship is really two one-to-many relationships with a third table whose primary key consists of two fields—the [foreign keys](#) from the two other tables.

You can also create a relationship between a table and itself. This is useful in situations where you need to perform a Lookup within the same table. In the Employees table, for example, you can define a relationship between the EmployeeID and ReportsTo fields, so that the ReportsTo field can display employee data from a matching EmployeeID.

Note If you drag a field that isn't a primary key and doesn't have a unique index to another field that isn't a primary key and doesn't have a unique index, an indeterminate relationship is created. In queries containing tables with an indeterminate relationship, Microsoft Access displays a default [join](#) line between the tables, but [referential integrity](#) won't be enforced, and there's no guarantee that records are unique in either table.

[Referential integrity](#)

Referential integrity is a system of rules that Microsoft Access uses to ensure that relationships between records in related tables are valid, and that you don't accidentally delete or change related data. You can set referential integrity when all of the following conditions are met:

- The matching field from the [primary table](#) is a [primary key](#) or has a [unique index](#).

- The related fields have the same [data type](#). There are two exceptions. An [AutoNumber](#) field can be related to a Number field with a **FieldSize** property setting of **Long Integer**, and an AutoNumber field with a **FieldSize** property setting of **Replication ID** can be related to a Number field with a **FieldSize** property setting of **Replication ID**.
- Both tables belong to the same Microsoft Access database. If the tables are linked tables, they must be tables in Microsoft Access format, and you must open the database in which they are stored to set referential integrity. Referential integrity can't be enforced for linked tables from databases in other formats.

The following rules apply when you use referential integrity:

- You can't enter a value in the [foreign key](#) field of the related table that doesn't exist in the primary key of the primary table. However, you can enter a [Null](#) value in the foreign key, specifying that the records are unrelated. For example, you can't have an order that is assigned to a customer that doesn't exist, but you can have an order that is assigned to no one by entering a **Null** value in the CustomerID field.
- You can't delete a record from a primary table if matching records exist in a related table. For example, you can't delete an employee record from the Employees table if there are orders assigned to the employee in the Orders table.
- You can't change a primary key value in the primary table, if that record has related records. For example, you can't change an employee's ID in the Employees table if there are orders assigned to that employee in the Orders table.

[Cascading updates and deletes](#)

For relationships in which [referential integrity](#) is enforced, you can specify whether you want Microsoft Access to automatically [cascade update](#) and [cascade delete](#) related records. If you set these options, delete and update operations that would normally be prevented by referential integrity rules are allowed. When you delete records or change [primary key](#) values in a [primary table](#), Microsoft Access makes necessary changes to related tables to preserve referential integrity.

If you select the **Cascade Update Related Fields** check box when defining a

relationship, any time you change the primary key of a record in the primary table, Microsoft Access automatically updates the primary key to the new value in all related records. For example, if you change a customer's ID in the Customers table, the CustomerID field in the Orders table is automatically updated for every one of that customer's orders so that the relationship isn't broken. Microsoft Access cascades updates without displaying any message.

Note If the primary key in the primary table is an [AutoNumber](#) field, setting the **Cascade Update Related Fields** check box will have no effect, because you can't change the value in an AutoNumber field.

If you select the **Cascade Delete Related Records** check box when defining a relationship, any time you delete records in the primary table, Microsoft Access automatically deletes related records in the related table. For example, if you delete a customer record from the Customers table, all the customer's orders are automatically deleted from the Orders table (this includes records in the Order Details table related to the Orders records). When you delete records from a form or datasheet with the **Cascade Delete Related Records** check box selected, Microsoft Access warns you that related records may also be deleted. However, when you delete records using a [delete query](#), Microsoft Access automatically deletes the records in related tables without displaying a warning.

[Show All](#)

About table relationships (ADP)

You can create relationships between your tables in a database diagram to show how the columns in one table are linked to columns in another table.

[What are table relationships?](#)

In a relational database, relationships enable you to prevent redundant data. For example, if you are designing a database that will track information about books, you might have a table called `titles` that stores information about each book, such as the book's title, date of publication, and publisher. There is also information you might want to store about the publisher, such as the publisher's phone number, address, and zip code. If you were to store all of this information in the `titles` table, the publisher's phone number would be duplicated for each title that the publisher prints.

A better solution is to store the publisher information only once in a separate table, `publishers`. You would then put a pointer in the `titles` table that references an entry in the publisher table.

To make sure your data is not out of sync, you can enforce referential integrity between the `titles` and `publishers` tables. Referential integrity relationships help ensure information in one table matches information in another. For example, each title in the `titles` table must be associated with a specific publisher in the `publishers` table. A title cannot be added to the database for a publisher that does not exist in the database.

[Types of table relationships](#)

A relationship works by matching data in key columns— usually columns with the same name in both tables. In most cases, the relationship matches the primary key from one table, which provides a unique identifier for each row, with an entry in the foreign key in the other table. For example, sales can be associated with the specific titles sold by creating a relationship between the `title_id` column in the `titles` table (the primary key) and the `title_id` column in the `sales` table (the foreign key).

There are three types of relationships between tables. The type of relationship that is created depends on how the related columns are defined.

[One-to-many relationships](#)

A one-to-many relationship is the most common type of relationship. In this type of relationship, a row in table A can have many matching rows in table B, but a row in table B can have only one matching row in table A. For example, the `publishers` and `titles` tables have a one-to-many relationship: each publisher produces many titles, but each title comes from only one publisher.

A one-to-many relationship is created if only one of the related columns is a primary key or has a unique constraint.

The primary key side of a one-to-many relationship is denoted by a key  symbol. The foreign key side of a relationship is denoted by an infinity  symbol.

[Many-to-many relationships](#)

In a many-to-many relationship, a row in table A can have many matching rows in table B, and vice versa. You create such a relationship by defining a third table, called a junction table, whose primary key consists of the foreign keys from both table A and table B. For example, the `authors` table and the `titles` table have a many-to-many relationship that is defined by a one-to-many relationship from each of these tables to the `titleauthors` table. The primary key of the `titleauthors` table is the combination of the `au_id` column (the `authors` table's primary key) and the `title_id` column (the `titles` table's primary key).

[One-to-one relationships](#)

In a one-to-one relationship, a row in table A can have no more than one matching row in table B, and vice versa. A one-to-one relationship is created if both of the related columns are primary keys or have unique constraints.

This type of relationship is not common because most information related in this way would be all in one table. You might use a one-to-one relationship to:

- Divide a table with many columns.
- Isolate part of a table for security reasons.
- Store data that is short-lived and could be easily deleted by simply deleting the table.
- Store information that applies only to a subset of the main table.

The primary key side of a one-to-one relationship is denoted by a key  symbol. The foreign key side is also denoted by a key  symbol.

[Overview of referential integrity](#)

[What is referential integrity?](#)

Referential integrity is a system of rules that ensure relationships between rows in related tables are valid and that you do not accidentally delete or change related data.

When referential integrity is enforced, you must observe the following rules:

- You cannot enter a value in the foreign key column of the related table if that value does not exist in the primary key of the related table. However, you can enter a null in the foreign key column. For example, you cannot indicate that a job is assigned to an employee who is not included in the employee table, but you can indicate that an employee has no assigned job by entering a null in the `job_id` column of the `employee` table.
- You cannot delete a row from a primary key table if rows matching it exist in a related table. For example, you cannot delete a row from the `jobs` table if there are employees assigned to the job represented by that row in the `employee` table.
- You cannot change a primary key value in the primary key table if that row has related rows. For example, you cannot delete an employee from the `employee` table if that employee is assigned to a job in the `jobs` table.

You can set referential integrity when all of the following conditions are met:

- The matching column from the primary table is a primary key or has a unique constraint.
- The related columns have the same data type and size.
- Both tables belong to the same database.

[Enforced and unenforced relationships in database diagrams](#)

Creating a relationship line in a database diagram automatically enforces referential integrity by creating a foreign key constraint on the related table. An enforced relationship appears in your database diagram as a solid line. For example:



An unenforced relationship, whose foreign key constraint is disabled, appears in your diagram as a dashed line. For example:



Depending on the features of your database, you can set options to disable the foreign key constraint for certain conditions, for example, during INSERT and UPDATE transactions.

[Show All](#)

Define relationships between tables (MDB)

When you create a [relationship](#) between tables, the related fields don't have to have the same names. However, related fields must have the same [data type](#) unless the [primary key](#) field is an [AutoNumber](#) field. You can match an AutoNumber field with a Number field only if the **FieldSize** property of both of the matching fields is the same. For example, you can match an AutoNumber field and a Number field if the **FieldSize** property of both fields is **Long Integer**. Even when both matching fields are Number fields, they must have the same **FieldSize** property setting.



[Define a one-to-many or a one-to-one relationship](#)

1. Close any tables you have open. You can't create or modify relationships between open tables.
2. Press F11 to switch to the [Database window](#).
3. Click **Relationships**  on the toolbar.
4. If you haven't yet defined any relationships in your database, the **Show Table** dialog box is automatically displayed.

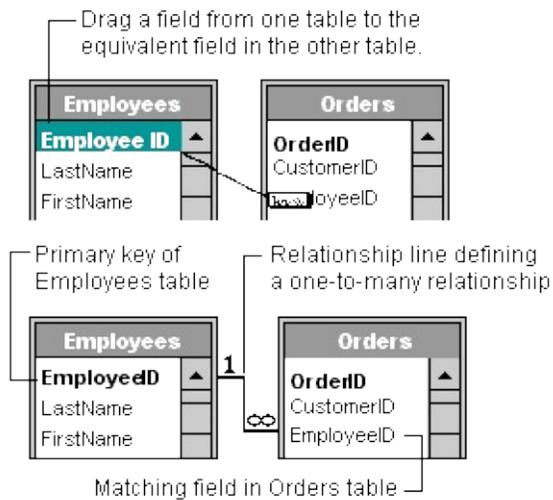
If you need to add the tables you want to relate and the **Show Table** dialog box isn't displayed, click **Show Table**  on the toolbar.

5. Double-click the names of the tables you want to relate, and then close the **Show Table** dialog box. To create a relationship between a table and itself, add that table twice.
6. Drag the field that you want to relate from one table to the related field in the other table.

To drag multiple fields, press the CTRL key, click each field, and then drag them.

In most cases, you drag the primary key field (which is displayed in bold text) from one table to a similar field (often with the same name) called the

[foreign key](#) in the other table.



7. The **Edit Relationships** dialog box is displayed. Check the field names displayed in the two columns to ensure they are correct. You can change them if necessary.

Set the relationship options if necessary.

8. Click the **Create** button to create the relationship.
9. Repeat steps 5 through 8 for each pair of tables you want to relate.

When you close the [Relationships window](#), Microsoft Access asks if you want to save the layout. Whether you save the layout or not, the relationships you create are saved in the database.

Note You can create relationships using queries as well as tables. However, [referential integrity](#) isn't enforced with queries.

[Define a many-to-many relationship](#)

1. Create the two tables that will have a [many-to-many relationship](#).
2. Create a third table, called a junction table, and add to the junction table new fields with the same definitions as the primary key fields from each of the other two tables. In the junction table, the primary key fields function as [foreign keys](#). You can add other fields to the junction table, just as you can to any other table.
3. In the junction table, set the primary key to include the primary key fields

from the other two tables. For example, in an Order Details junction table, the primary key would be made up of the OrderID and ProductID fields.

 [How?](#)

1. Open a table in [Design view](#).
2. Select the field or fields you want to define as the [primary key](#).

To select one field, click the [row selector](#) for the desired field.

To select multiple fields, hold down the CTRL key and then click the row selector for each field.

3. Click **Primary Key**  on the toolbar.

Note If you want the order of the fields in a multiple-field primary key to be different from the order of those fields in the table, click **Indexes**  on the toolbar to display the [Indexes window](#), and then reorder the field names for the [index](#) named PrimaryKey.

4. Define a [one-to-many relationship](#) between each of the two [primary tables](#) and the junction table.

[Show All](#)

Create a relationship between tables (ADP)

You create a relationship between two tables when you want to associate rows of one table with rows of another.

[Create a relationship in a database diagram](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector  for the database column or combination of columns that you want to relate to a column in another table.
3. While the pointer is positioned over the row selector, click and drag to the related table.
4. Release the mouse button. The **Create Relationship** dialog box appears and attempts to match the columns you selected with columns of the same name and data type in the related table.
5. In the **Create Relationship** dialog box, confirm that the columns you want to relate are shown in the **Primary key table** and **Foreign key table** lists.
6. Choose **OK** to create the relationship.

On the diagram, the primary key side of the relationship is denoted by a key  symbol. In one-to-one relationships, the table that initiated the relationship determines the primary key side. For example, if you create a relationship from the `pub_id` column in the `publishers` table to the `pub_id` column in the `pub_info` table, then the `publishers` table is on the primary key side of the relationship.

[Create a relationship in Table Designer](#)

1. Click **Tables**  under **Objects**, click the table that is on the foreign key side of the relationship, and then click **Design** on the Database window toolbar.

2. Right-click in the Table Designer and choose **Relationships**.
3. Click the **New** button.
4. From the drop-down list in **Primary key table**, choose the table that will be on the primary-key side of the relationship. In the grid beneath, enter the columns contributing to the table's primary key. In the adjacent grid cell to the left of each column, enter the corresponding foreign-key column of the foreign-key table.

The table designer suggests a name for the relationship. To change this name, edit the contents of the **Relationship name** text box.

5. Choose **Close** to create the relationship.

[Show All](#)

Delete a relationship (MDB)

1. Close any tables you have open. You can't delete [relationships](#) between open tables.
2. Press F11 to switch to the [Database window](#).
3. Click **Relationships**  on the toolbar.
4. If the tables whose relationship you want to delete aren't displayed, click **Show Table**  on the toolbar and double-click each table you want to add. Then click **Close**.
5. Click the relationship line for the relationship you want to delete (the line will turn bold when it's selected), and then press the DELETE key.

[Show All](#)

Set or change the join type (MDB)

[Define the default join type for a relationship between two tables](#)

Defining the [join](#) type for a [relationship](#) in the [Relationships window](#) doesn't affect the relationship itself; it sets the kind of join that will be used by default when creating queries based on the related tables in a [Microsoft Access database](#). You can always override the default join type later when defining a query.

1. Close both of the related tables.
2. Press F11 to switch to the [Database window](#).
3. Click **Relationships**  on the toolbar to open the Relationships window.
4. Double-click the middle section of a join line between two tables to open the **Relationships** dialog box.
5. Click the **Join Type** button, and then click the desired join type.

Option 1 defines an [inner join](#). This is the default.

Option 2 defines a [left outer join](#).

Option 3 defines a [right outer join](#).

Note The **Join Type** button won't be enabled if the tables are linked tables. If the linked tables are in Microsoft Access format, you can open the database in which they are stored to set the join type.

[Change the join type in a query](#)

1. Open a query in [Design view](#).
2. Double-click the join line between the [field lists](#) for the tables or queries.



 Join line

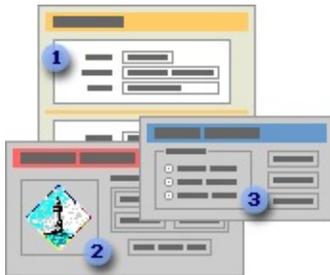
3. In the **Join Properties** dialog box, click the join option you want, and then click **OK**.

Note From the [Relationships window](#), you can change the default join type for tables.

[Show All](#)

About forms

A form is a type of a database object that is primarily used to enter or display data in a database. You can also use a form as a switchboard that opens other forms and reports in the database, or as a custom dialog box that accepts user input and carries out an action based on the input.



- 1 A data-entry form
- 2 A switchboard form
- 3 A custom dialog box

Most forms are bound to one or more tables and [queries](#) in the database. A form's [record source](#) refers to the fields in the underlying tables and queries. A form need not contain all the fields from each of the tables or queries that it is based on.

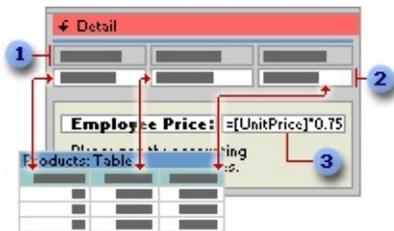
A bound form stores or retrieves data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the form's design.



- 1 Graphic elements, such as lines and rectangles, are stored in the form's design.

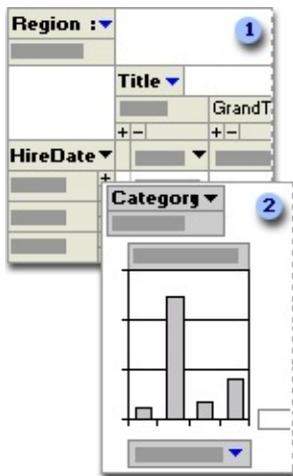
- 2 Data comes from the fields in the underlying record source.
- 3 A calculation comes from an expression, which is stored in the form's design.
- 4 Descriptive text is stored in the form's design.

You create a link between a form and its record source by using graphical objects called [controls](#). The most common type of control used to display and enter data is a text box.



- 1 Labels display descriptive text.
- 2 Text boxes display data from and enter data into the Products table.
- 3 A text box uses an expression to calculate a total.

You can also open a form in PivotTable view or PivotChart view to analyze data. In these views, you can dynamically change the layout of a form to present data in different ways. You can rearrange row headings, column headings, and filter fields until you achieve the desired layout. Each time you change the layout, the form immediately recalculates the data based on the new arrangement.



- 1 In PivotTable view, you can view detail or summarized data by arranging fields in the filter, row, column, and detail areas.
- 2 In PivotChart view, you can display data visually by selecting a chart type and viewing data by arranging fields in the filter, series, category, and data areas.

[Creating a form](#)

You can create a form quickly by using the AutoForm command or a wizard. AutoForm creates a form that displays all fields and records in the underlying table or [query](#). A wizard asks you questions and creates a form based on your answers. You can then customize the form the way you want it in [Design view](#).

[Customizing a form](#)

[In Design view](#)

You can customize a form in Design view in the following ways:

Record source Change the tables and queries that a form is based on.

Controlling and assisting the user You can set form properties to allow or prevent users from adding, deleting, or editing records displayed in a form. You can also add custom Help to a form to assist your users with using the form.

Form window You can add or remove **Maximize** and **Minimize** buttons, short cut menus, and other Form window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a form. You can also set section properties to control the appearance and printing of a form.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a form.

[In PivotTable or PivotChart view](#)

You can customize a form in PivotTable or PivotChart view in the following

ways:

Add, move, or remove fields You can add fields to the filter, row, column, and detail areas in PivotTable view, and to the filter, category, series, and data areas in PivotChart view. You can also move fields from one area to another and remove fields from the view.

Filter records You can filter data displayed in the view by adding or moving a field to the filter area. You can also filter a field in the row and column area.

Sort records You can sort items in row or column fields in ascending or descending order. You can also sort items in custom order in PivotTable view.

Group records You can group items in row or column fields on intervals, or create custom groups.

Format elements and change captions In PivotTable view, you can change the font settings, alignment, background color, and number format of a field. You can also change the captions of fields and custom groups. In PivotChart view, you can change the chart type, format data markers, and more.

[Show All](#)

Ways to work with data in a form

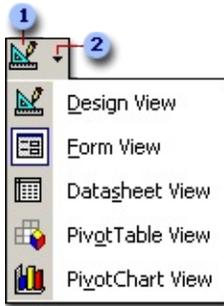
In [Form view](#), you have all the tools you need to work with your data.



- 1 Switch views
- 2 Edit, check spelling, or print
- 3 Sort, filter, or find records
- 4 Add or delete records
- 5 Navigate through records

[Changing to other views](#)

You can change to other views of a form using the **View** button on the toolbar. The **View** button is available in all form views.



- 1 Clicking the button will change to the view indicated by the button graphic.
- 2 The arrow next to the button shows a list of possible views to choose from.

[Sorting, filtering, and finding data](#)

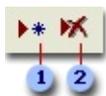
You can sort the records in a form by the data in one field.



- 1 Sort records in ascending or descending order.
- 2 Click one of the filter buttons to limit the records you see in your form.
- 3 Search and replace values in records.

[Adding and deleting records](#)

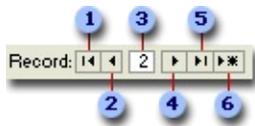
You can add and delete records in Form view.



- 1 This button displays a blank record for you to fill in.
- 2 This button deletes the current record.

[Navigating records](#)

You can move between the records displayed in a form in Form view.



- 1 First record
- 2 Previous record
- 3 Type a record number to move to.
- 4 Next record
- 5 Last record
- 6 Blank (new) record

[Show All](#)

Troubleshoot forms

[Viewing and editing data](#)

[My form is blank.](#)

A form may appear blank for any of the following reasons:

- The form may not be bound to an underlying [record source](#). To bind a form to an underlying record source, open the form in [Design view](#), and then double-click the [form selector](#) to open the property sheet. In the **RecordSource** property box, click a record source in the list or, if the record source is a [query](#), click the **Build** button  to open the Query Builder.
- The form may be bound to an underlying record source that doesn't have any data. Check the underlying record source to make sure that it has data.
- The form may be based on a query that doesn't return any data. To see if the query returns data, double-click the form selector to open the form's property sheet, click the **Build** button next to the **RecordSource** property box to open the Query Builder, and then click **View**  in the Query Builder.
- The form may be open in Data Entry mode. When a form is open in Data Entry mode, it displays a blank record so that you can add data. To show all the records, click the **Remove Filter/Sort** command on the **Records** menu.
- The [form header](#) or [form footer](#) may be so large that there's no room for the [detail section](#).
- When you open a form for the first time in PivotTable or PivotChart view, you don't see any data. Using the field list, add fields to the different areas of the view.

[I can see data from only one table.](#)

To see data from more than one table, you must base your form on a [query](#) that brings the data from the tables together. To create a multiple-table query without closing the form, do the following:

1. In form [Design view](#), double-click the [form selector](#) to open the form's property sheet.
2. Click in the **RecordSource** property box, and then click the **Build** button  next to the property box to open the Query Builder.
3. In the Query Builder, add the tables and fields you want to include in the underlying query, and then close the Query Builder.

Microsoft Access updates the **RecordSource** property to use the query you created.

Note You can also create a multiple-table form by using the Form Wizard, PivotTable Wizard, or PivotChart Wizard.

 [I can't edit data in my form.](#)

You may not be able to edit data in a form for any of the following reasons:

- The form was created as a read-only form. If the **AllowDeletions**, **AllowAdditions**, and **AllowEdits** properties of the form are set to **No**, you won't be able to make changes to the underlying data.
- The **Locked** property for one or more [controls](#) may be set to **Yes**.
- Someone else may be using the form at the same time, and the **RecordLocks** property for the form is set to **All Records** or **Edited Record**. You'll see the locked record indicator  in the [record selector](#) if this is the case. Wait a while, and try again.
- You may be trying to edit data in a calculated control. A calculated control displays the result of an [expression](#). The data displayed in a calculated control isn't stored, so it can't be edited.
- The underlying [query](#) or SQL statement for the form may not be updatable.

- You can't edit data in PivotTable or PivotChart view.

[PivotTable forms](#)

Note A PivotTable form is different from a form that is opened in PivotTable view. You cannot create PivotTable forms in Microsoft Access 2002 or later. Instead, you can open an existing form in PivotTable view and build the layout of the view, or use the PivotTable wizard to help you design the PivotTable view for a new form.

[Some of the data in the PivotTable form is cut off and there aren't any scroll bars to see the rest of the data.](#)

- To scroll through the data in the PivotTable form, click the **Edit PivotTable Object** button, which opens Microsoft Excel. You can then see the entire PivotTable form and scroll through the data.

[I clicked the button to edit the PivotTable form, but Microsoft Excel won't let me make any modifications.](#)

Before you can modify a PivotTable form, you need to update the data that's displayed in it.

1. Open the PivotTable form in [Form view](#).
2. Click the **Edit PivotTable Object** button.
3. On the **Data** menu in Microsoft Excel, click **Refresh Data**.

Note When you're editing the PivotTable form, you are using Microsoft Excel and can get Excel Help on PivotTable reports. If you don't find the answer you need in the Excel Help topics, you can use Microsoft Access Help when the PivotTable form is open in Form view, in Access.

[If I open Microsoft Excel without going through the form containing the PivotTable object, I can't find the PivotTable form.](#)

A PivotTable object is an embedded object that is stored in Microsoft Access. If you want to work with the PivotTable object by using Excel's PivotTable Report

Wizard, click the **Edit PivotTable Object** button on the Access form. Once you have the PivotTable object open in Excel, you can save a copy of it as an Excel PivotTable report. However, the PivotTable report in Excel and the PivotTable object on the Access form aren't linked. If you change the PivotTable object in Access, the changes won't be reflected in the PivotTable report in Excel, and vice versa, so the data will be inconsistent.

[My PivotTable form looks different from a PivotTable form that is created using Microsoft Access 2002 or later](#)

You cannot create PivotTable forms in Access 2002 or later. Instead, you can open a form in PivotTable view or PivotChart view to get the same functionality as a PivotTable form. You can also use the PivotTable or Chart wizards to build the view for a new form.

[Subforms](#)

[All of my records show up in my subform - they don't change when I move from record to record in the main form.](#)

You need to link your main form and subform using the **LinkChildFields** and **LinkMasterFields** properties.

1. Open the main form in [Design view](#).
2. Make sure the [subform control](#) is selected, and then click **Properties**  on the toolbar to display the property sheet for the subform control.
3. Type the linking fields or controls in the **LinkChildFields** and **LinkMasterFields** properties. To enter more than one linking field or control, separate the field or control names with a semicolon. If you enter more than one field or control name, you must enter the names in the same order in both properties.
 - In the **LinkChildFields** property box, enter the name of each linking field in the subform. (You can't use the name of a control in this property.)
 - In the **LinkMasterFields** property box, enter the name of each linking

field or control in the main form.

If you're unsure what the linking field or control is for either property, click the **Build** button next to the property box to open the **Subform Field Linker**.

4. Switch to [Form view](#) to test the link.

If you've already set the **LinkChildFields** and **LinkMasterFields** properties and the subform still doesn't work, make sure that you aren't using fields that have been renamed or deleted. Also make sure that you've spelled the field names correctly.

Notes

- The linking fields don't have to appear in the main form or the subform, but they must be included in the underlying record source.
- The linking fields must contain the same kind of data and have the same or a compatible [data type](#) or field size.
- Make sure that the **LinkChildFields** property is set to the name of a field (as specified in the **ControlSource** property), not the name of a control (as specified in the **Name** property). The **LinkMasterFields** property can be set to the name of either a field or a control.

[I get #Error? or #Name? when I try to display data from my subform on my main form.](#)

To display a value from a subform on a main form, create a [calculated control](#) that refers to the subform control using the following syntax:

```
=Forms![formname]![subformname]![controlname]
```

For example, suppose you have an Orders form and an Orders subform. In the form footer of the subform, you have a text box named **OrderSubtotal** that sums the records in the subform. To display the value of this control on the main form, use the following expression:

```
=Forms![Orders]![Orders Subform]![OrderSubtotal]
```

If you're already using this syntax and you still get an error, check to make sure that the form, subform, and control names are correct. Make sure that the names you're referring to are the names specified in the **Name** property for each object. Also make sure that you're including the exclamation point in the right places.

[I can't display form headers or form footers in my subform datasheet.](#)

A form or subform viewed in [Datasheet view](#) doesn't display a form header and footer even if you've included them in [Design view](#). You can display a form header and footer in a subform and make it look like a [datasheet](#) by sizing the [detail section](#) of the form to the size of a single row and setting the **DefaultView** property of the subform to **Continuous Forms**.

[Property sheet](#)

[I can't open the property sheet in Form view or Datasheet view.](#)

To be able to open the property sheet when a form is open in [Form view](#) or [Datasheet view](#), the **AllowDesignChanges** property of the form must be set to **All Views**.

[Images](#)

[I get an error message when I try to add an image in JPEG graphic format.](#)

Do one of the following:

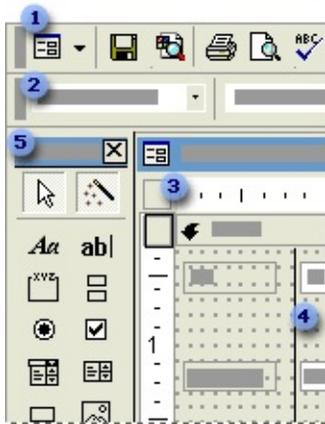
- Convert the image to GIF or BMP graphic format.
- Reduce the size of the image.

[Show All](#)

About views of a form

[Design view](#)

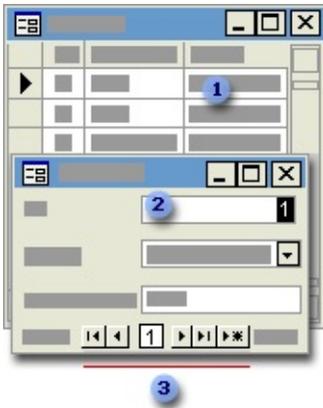
To create a form, you work in Design view. Viewing a form in Design view is like sitting at a workbench surrounded by useful tools.



- 1 Click the **View** button to switch to another view.
- 2 Use the **Formatting** toolbar to change the fonts or font sizes, align text, change border or line widths, or apply colors or special effects.
- 3 Align [controls](#) with the rulers.
- 4 Place controls in the [detail section](#), or one of the other form sections.
- 5 Use the toolbox to add controls, such as labels and text boxes.

[Form view and Datasheet view](#)

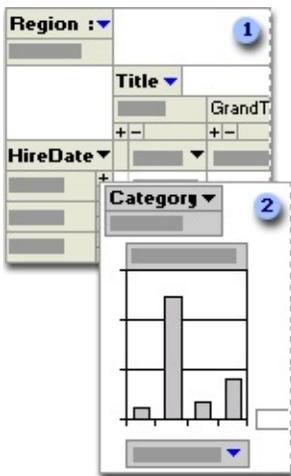
After you've created a form in Design view, you can view it in Form view or Datasheet view.



- 1 In Datasheet view, you can view records in row and column format, so you can see many records at one time.
- 2 In Form view, you can typically look at only one record at a time.
- 3 In Form view or Datasheet view, use the [navigation buttons](#) to quickly navigate between records.

[PivotTable view and PivotChart view](#)

You can open a form that is bound to data in PivotTable view or PivotChart view. In these views, you can dynamically change the layout of a form to analyze data in different ways. You can rearrange row headings, column headings, and filter fields until you achieve the desired layout. Each time you change the layout, the form immediately recalculates the data based on the new arrangement.



- 1 In PivotTable view, you can view detail or summarized data by arranging fields in the filter, row, column, and detail areas.
- 2 In PivotChart view, you can display data visually by selecting a chart type and viewing data by arranging fields in the filter, series, category, and data areas.

[Show All](#)

Create a form

There are three ways to create a form.

Based on a single table or query by using AutoForm. AutoForm creates a form that displays all fields and records in the underlying table or [query](#). If the [record source](#) you select has related tables or queries, the form will also include all the fields and records from those record sources.

[How?](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, click one of the following wizards:
 - **AutoForm: Columnar** Each field appears on a separate line with a label to its left.
 - **AutoForm: Tabular** The fields in each record appear on one line, with the labels displayed once at the top of the form.
 - **AutoForm: Datasheet** The fields in each record appear in row-and-column format, with one record in each row and one field in each column. The field names appear at the top of each column.
 - **AutoForm: PivotTable** The form opens in PivotTable view. You can add fields by dragging them from the field list to the different areas in the view.
 - **AutoForm: PivotChart** The form opens in PivotChart view. You can add fields by dragging them from the field list to the different areas in the view.
4. Click the table or query that includes the data you want to base your form on.
5. Click **OK**.

Microsoft Access applies the last [autoformat](#) you used to the form. If you haven't created a form with a wizard before or haven't used the **AutoFormat** command on the **Format** menu, it uses the Standard autoformat.

[Tip](#)

You can also create a columnar AutoForm based on the open record source or the record source that's selected in the Database window. Click **AutoForm** on the **Insert** menu, or click the arrow next to the **New Object** button on the toolbar, and then click **AutoForm**. If the record source you select has related tables, AutoForm adds a subform that contains the related tables displayed as [datasheets](#) and [subdatasheets](#).

Based on one or more table or query with a wizard. The wizard asks you detailed questions about the record sources, fields, layout, and format you want and creates a form based on your answers.

How?

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, click the wizard that you want to use. A description of the wizard appears in the left side of the dialog box.
4. Click the name of the table or other record source that includes the data you want to base your form on.

Note You don't need to do this step if you selected Form Wizard or PivotTable Wizard in step 3— you can specify the record source for the form later.

5. Click **OK**.
6. Follow the instructions in the wizard.

If the resulting form doesn't look the way you want, you can change it in [Design view](#), PivotTable view, or PivotChart view.

Notes

- If you want to include fields from multiple tables and queries in your form, don't click **Next** or **Finish** after selecting the fields from the first table or query in the Form Wizard. Repeat the steps to select a table or query, and pick fields that you want to include in the form, until you have selected all the required fields.
- If you click one of the **AutoForm** options, Microsoft Access uses the

[autoformat](#) you last specified, either in the Form Wizard or using the **AutoFormat** command on the **Format** menu in Design view.

On your own in Design view. You create a basic form and customize it in [Design view](#) to suit your requirements.

[How?](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, click **Design View**.
4. Click the name of the table or other [record source](#) that includes the data you want to base your form on. If the form won't contain data (for example, if you want to create a form to use as a switchboard to open other forms or reports, or if you want to create a custom dialog box), don't select anything from this list.

[Tip](#)

If you want to create a form that uses data from more than one table, base your form on a [query](#).

5. Click **OK**.

Microsoft Access displays the form in [Design view](#).

[Show All](#)

About using a template to create a form or report

When you create a form or report without using a wizard, Microsoft Access uses a template to define the default characteristics of the form or report.

The template determines which sections a form or report will have and defines each section's dimensions. The template also contains all the default property settings for the form or report and its sections and [controls](#). However, a template doesn't create controls on a new form or report.

The default templates for forms and reports are called Normal. However, you can use any existing form or report as a template. You can also create a form or report to use only as a template. Changing the template doesn't have any effect on existing forms or reports.

Microsoft Access saves the settings for the **Form Template** and **Report Template** options in your Microsoft Access [workgroup information file](#), not in your [Microsoft Access database](#) (the .mdb file) or [Microsoft Access project](#) (the .adp file). When you change an option setting, the change applies to any Access database or Access project you open or create.

If your templates are not in an Access database or Access project, Microsoft Access uses the Normal template for any new forms and reports you create. However, the names of your templates appear in the **Form Template** and **Report Template** options in every Access database or Access project in your database system, even if the templates are not in every Access database or Access project.

Set the default template for forms and reports

You can either use an existing form or report as a template, or create a form or report specifically for use as a template. Changing the default template does not have any effect on existing forms and reports.

1. On the **Tools** menu, click **Options**.
2. Click the **Forms/Reports** tab.
3. Type the name of the new template in the **Form Template** or **Report Template** box.
4. Click **OK**.

Note To see the name of the template that is currently used for new forms or reports, click **Options** on the **Tools** menu, and then click the **Forms/Reports** tab.

[Show All](#)

Use a predefined format

1. Open a form or a report in [Design view](#).
2. Do one of the following:

To format the entire form or report, click the [form selector](#) or the [report selector](#).

To format an individual section, click the [section selector](#).

To format one or more [controls](#), select the controls.

3. Click **AutoFormat**  on the toolbar.
4. Click one of the formats in the list.
5. If you want to specify which attributes to apply (font, color, or border), click **Options**.

Note You must select the entire form or report to apply a background picture.

[Show All](#)

Create, update, or delete an autoforamt

1. Open a form or report in [Design view](#).
2. On the **Format** menu, click **AutoFormat**.
3. In the **AutoFormat** dialog box, click an autoforamt in the list.
4. Click the **Options** button, and then make sure the attributes you want to modify are selected.
5. Click the **Customize** button, and then click a customization option. You can do one of the following:

Create a new autoforamt based on the open form or report.

Update the selected autoforamt based on the open form or report.

Delete the selected autoforamt from the list.

6. Click **OK** twice to close the dialog boxes.

[Show All](#)

Bind a blank form or report to one or more tables or queries

1. Set the [record source](#) of the form or report.

[How?](#)

After you have created a form or report, you can modify its source of data to include or exclude one or more fields, columns, tables, or [queries](#). Changing the record source will change the contents of the [field list](#).

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
3. Do one of the following:
 - To include all fields or columns from a single table or query in the record source, click the arrow in the **RecordSource** property box to select from a list of tables and queries.
 - To base the form or report on certain fields or columns, or on more than one table or query, click the **Build** button next to the **RecordSource** property to open the query.
2. Use the [field list](#) to add [controls](#) bound to the underlying fields or columns to the form or report.

[Show All](#)

Ways to customize a form

[In Design view](#)

You can customize a form in Design view in the following ways:

Record source Change the tables and queries that a form is based on.

Controlling and assisting the user You can set form properties to allow or prevent users from adding, deleting, or editing records displayed in a form. You can also add custom Help to a form to assist your users with using the form.

Form window You can add or remove **Maximize** and **Minimize** buttons, short cut menus, and other Form window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a form. You can also set section properties to control the appearance and printing of a form.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a form.

[In PivotTable or PivotChart view](#)

You can customize a form in PivotTable or PivotChart view in the following ways:

Add, move, or remove fields You can add fields to the filter, row, column, and detail areas in PivotTable view, and to the filter, category, series, and data areas in PivotChart view. You can also move fields from one area to another and remove fields from the view.

Filter records You can filter data displayed in the view by adding or moving a field to the filter area. You can also filter a field in the row and column area.

Sort records You can sort items in row or column fields in ascending or

descending order. You can also sort items in custom order in PivotTable view.

Group records You can group items in row or column fields on intervals, or create custom groups.

Format elements and change captions In PivotTable view, you can change the font settings, alignment, background color, and number format of a field. You can also change the captions of fields and custom groups. In PivotChart view, you can change the chart type, format data markers, and more.

[Show All](#)

Change the record source or connection information

[Change the record source of a form or report](#)

After you have created a form or report, you can modify its source of data to include or exclude one or more fields, columns, tables, or [queries](#). Changing the record source will change the contents of the [field list](#).

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
3. Do one of the following:
 - To include all fields or columns from a single table or query in the record source, click the arrow in the **RecordSource** property box to select from a list of tables and queries.
 - To base the form or report on certain fields or columns, or on more than one table or query, click the **Build** button next to the **RecordSource** property to open the query.

[Change the connection information of a data access page](#)

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. On the property sheet, click the **Data** tab.
4. Do one of the following:

[Set or edit the **ConnectionString** property of the page](#)

Note Editing the **ConnectionString** property of a page that is linked to a connection file will break the link between the page and the connection file.

- In the **ConnectionString** property box, either type the new connection string or click the **Build** button to use the **Data Link Properties** dialog box to build the connection string.



Tip

If you need more room to type the connection string, press SHIFT+F2 to open the **Zoom** box.



[Specify or change the connection file that the page is linked to](#)

- In the **ConnectionFile** property box, either type the name of the new connection file or click the **Build** button  to locate the file.

Note Microsoft Access will automatically update the **ConnectionString** property to reflect the new connection information.



[Edit the connection file of the page](#)

Note If the page is linked to a connection file, you can edit the contents of the file. Your edits will affect other pages that are linked to the connection file.

1. In the **ConnectionFile** property box, click the **Build** button .
2. In the **Select Data Source** dialog box, right-click the connection file, point to **Open With**, and click **Notepad**.
3. Search for "<odc:ConnectionString>" (without the quotation marks) to locate the connection information and make the changes you want.
4. Save and close the file.
5. In the **Select Data Source** dialog box, click **OK**.

Notes

- If you have specified a default connection file on the **Pages** tab in the **Options** dialog box, Microsoft Access will automatically set the **ConnectionFile** and **ConnectionString** properties for all new data access pages.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. For more information about a PivotTable list, spreadsheet, or chart, click **Help**  on that component's toolbar.

[Change the record source of a data access page section](#)

You can change the [record source](#) of a [section](#) on an ungrouped page, if the record source that you want to use contains all the fields that are already on the header section. For example, if you add fields from the Order Details table, and then decide to use the Order Details Extended query instead of the Order Details table, you can simply change the property of the header section.

1. Open the data access page in [Design view](#).
2. Double-click the [section bar](#) of the header section.
3. In the **RecordSource** property box, select a record source from the list.

Note The names of the sections and the **ControlSource** property of the controls will be automatically updated to reflect the new record source.

[Show All](#)

About creating a form with tabs or page breaks

You can create a form with multiple pages by adding either a [tab control](#) or a page break [control](#) to the form.

A form with tabs

A tab control is the easiest and most effective way to create a multiple-page form. With a tab control, the separate pages are all built into one control. To switch pages, you click one of the tabs.



- 1 General employee information is displayed on this page.
- 2 Personal information, such as home address and phone number, is displayed on this page.

A form with page breaks

You use a page break control to mark a horizontal break between controls on a form. A page break is active in [Form view](#) only when the **DefaultView** property of the form is set to **Single Form**.

In Form view, pressing the PAGE UP or PAGE DOWN keys will scroll the form to the page before or after the page break control. When a form is printed, a page break marks the start of a new page within a section.

[Show All](#)

Examples of expressions for page numbers

The following is a list of examples of page number expressions you can use in form or report [Design view](#) and the results that you will see in other views.

- Expression: =[Page]
Result: 1, 2, 3
- Expression: ="Page " & [Page]
Result: Page 1, Page 2, Page 3
- Expression: ="Page " & [Page] & " of " & [Pages]
Result: Page 1 of 3, Page 2 of 3, Page 3 of 3
- Expression: =[Page] & " of " & [Pages] & " Pages"
Result: 1 of 3 Pages, 2 of 3 Pages, 3 of 3 Pages
- Expression: =[Page] & "/"& [Pages] & " Pages"
Result: 1/3 Pages, 2/3 Pages, 3/3 Pages
- Expression: =[Country] & " - " & [Page]
Result: UK - 1, UK - 2, UK - 3
- Expression: =Format([Page], "000")
Result: 001, 002, 003

[Show All](#)

Add a tab or page break control to a form

1. Create a blank form.

 [How?](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, click **Design View**.
4. Click the name of the table or other [record source](#) that includes the data you want to base your form on. If the form won't contain data (for example, if you want to create a form to use as a switchboard to open other forms or reports, or if you want to create a custom dialog box), don't select anything from this list.

 [Tip](#)

If you want to create a form that uses data from more than one table, base your form on a query.

5. Click **OK**.

Microsoft Access displays the form in [Design view](#).

2. Do one of the following:

 [Add a tab control to the form](#)

1. Open a form in [Design view](#).
2. In the [toolbox](#), click the **Tab Control** tool  and then click the form where you want to place the control.

Microsoft Access adds a tab control with two pages. The first page is on top.

3. To add controls to the tab control, click the tab of the page you want to add controls to. Then add controls using any of the following methods:
 - Click **Field List**  on the [toolbar](#) to display the [field list](#) and then drag one or more fields to the tab page.
 - Click a tool in the toolbox and click on the tab page. You can add any type of control except another tab control.
 - Copy controls from another part of the form or from another page. (However, you can't drag controls from another part of a form or from another page.)
4. Customize the tab control.

 [How?](#)

In form [Design view](#), do one or more of the following:

1. 1. **Add, delete, or change the order of tabs**

- Right-click the border of the tab control, and then click **Insert Page, Delete Page, or Page Order**.

1. 1. **Change the tab order of controls on a page**

- Right-click the page, and then click **Tab Order**.

1. 1. **Specify whether the tab control can have more than one row of tabs**

- Double-click the edge of the tab control to display its property sheet, and then set the **MultiRow** property to **Yes**.

If the **MultiRow** property is set to **No**, Microsoft Access truncates tabs that don't fit within the width of the control and adds a scroll bar.

1. 1. **Specify whether to display tabs or command buttons at the top of the tab control**

- Double-click the edge of the tab control to display its property sheet, and then set the **Style** property to **Tabs**, **Buttons**, or **None**.

You might want to click **None** if you plan to use a command button outside of the tab control to determine which page of the tab control has [focus](#).

1. 1. **Set the height of the tabs**

- Double-click the edge of the tab control to display its property sheet, and then type a value in the **TabFixedHeight** box.

If the value is **0**, each tab will be high enough to fit its contents.

1. 1. **Set the width of the tabs**

- Double-click the edge of the tab control to display its property sheet, and then type a value in the **TabFixedWidth** box.

If the value is **0**, each tab will be wide enough to fit its contents and, if there is more than one row of tabs, the width of each tab will increase so that each row of tabs spans the width of the tab control. If the value is greater than zero, all tabs have the identical width specified by this property.

1. 1. **Specify the font properties of the captions on tabs**

- Double-click the edge of the tab control to display its property sheet, and then set the **FontName**, **FontSize**, **FontWeight**, **FontItalic**, or **FontUnderline** property.

You can't specify different settings for different tabs.

1. 1. **Specify the display text that appears on a tab**

- Double-click the tab for the page to display its property sheet, and then type the text in the **Caption** box.

If you don't specify a caption, Microsoft Access uses the text in the **Name** property box.

1. 1. **Add a picture to a tab**

- Double-click the tab for the page to display its property sheet, click the **Build** button (...) next to the **Picture** box, and then select a picture in the Picture Builder.

The picture appears to the left of the text specified in the **Caption** property. If you just want to display a picture and no text, enter a SPACEBAR space in the **Caption** property. Tab controls can contain bitmaps but not Microsoft Windows metafiles.

Note For more information about a property, click the property box in the property sheet and press F1.

2. Size the tab control as appropriate. Click each tab to make sure all the controls fit well within each tab.

Note Microsoft Access won't crop controls when you size the tab control. You may need to move controls to make the tab control smaller.

3. Switch to [Form view](#) to test the control.

[Add page breaks to the form](#)

1. Open the form in [Design view](#).
2. Click the **Page Break** tool  in the [toolbox](#).
3. Click where you want to place the page break. Place the page break above or below a [control](#) to avoid splitting data in that control.
4. Microsoft Access marks the page break on the left border of the form with a short dotted line.
5. **Note** If you want each page to be the same size and each window to show only one page at a time, design the form so that each page break is an equal distance from the other. You can do this by positioning the page break controls with the vertical ruler.

6. Double-click the [form selector](#) to open the property sheet, and click **Current Page** in the **Cycle** property box. This will prevent the user from moving to the next page by pressing the TAB key.
7. Remove the vertical scroll bar by setting the **ScrollBars** property to **Horizontal Only** or **Neither**.
8. Switch to [Form view](#) to test the form and size it so you can only see one page at a time. You can press the PAGE DOWN or PAGE UP keys to move between pages.

Note You might want to add a command button to each page that users can use to move the [focus](#) to the next or previous page.

[Show All](#)

Display a Web page on a form

Before you can add the Microsoft Web Browser [control](#) to a form, you must install Microsoft Internet Explorer.

1. Open the form in [Design view](#).
2. In the [toolbox](#), click **More Controls** .

A menu appears that lists all the registered [ActiveX controls](#) in your system.

3. On the menu of ActiveX controls, click **Microsoft Web Browser**.
4. On the form, click where you want to place the control.
5. Move and resize the control if necessary.

[Show All](#)

About themes

A theme is a set of unified design elements and color schemes for bullets, fonts, horizontal lines, background images, and other elements of a database object. Themes help you easily create professional and well-designed Microsoft Access applications.

Microsoft Office Access 2003 supports theming in two ways.



[Inherits the theme from the operating system](#)

If the operating system is Microsoft Windows XP SP 1 or later, and you have chosen a theme other than Windows Classic, Access will inherit the theme from the operating system. Most of the views, [controls](#), and dialog boxes will be customized to match the Windows theme. Exceptions are [reports](#), diagrams, SQL view of a [query](#), and [modules](#).

A control on a [form](#) will not inherit the Windows theme, if its property settings are different from what is listed below.

Property	Setting
	Sunken, Flat, or Etched.
Special Effect	Note If the value is set to Flat , the Border Style property must not be set to Transparent in order to inherit the theme. Solid or Transparent.
Border Style	Note If the value is set to Transparent , the Special Effect property must not be set to Flat in order to inherit the theme.
Border Color	0, -2147483640 (system color that corresponds to the default font color for dialogs) Hairline, 1, or 2
Border	Note The Border Width is ignored for check boxes and option

Width buttons.

You can prevent form controls from inheriting the Windows theme by clearing the **Use Windows Themed Controls on Forms** check box on the **Forms/Reports** tab in the **Options** dialog box.

Note This setting applies only to the current database or project.

The check box is selected by default in databases created using Office Access 2003.



[Provides special themes for data access pages](#)

For data access pages, you can choose from a list of themes that are installed with Access when a page is opened in [Design view](#).

When you apply a theme to a [data access page](#), the following elements are customized: body and heading styles, background color or graphic, table border color, horizontal lines, bullets, hyperlink colors, and controls. You can also select options to apply brighter colors to text and graphics, animate certain theme graphics, and apply a background to your data access page.

You can apply a theme to a page that does not have a theme, change the current theme, or remove a theme. Before applying a theme, you can preview a sample page that uses the theme.

Theme graphics are animated only when you view the data access page in a Web browser, not while you view the data access page in Microsoft Access.

A variety of themes are available on your hard drive when you install Access. If you have Microsoft FrontPage 4.0 or later installed, you can use the FrontPage themes as well. Additional themes are available from the World Wide Web. To download these themes, click **Microsoft Office Online** on the **Help** menu.

[Show All](#)

Apply, change, or remove a theme

[For forms](#)

[Apply the current Windows theme to form controls](#)

1. Make sure the operating system is Microsoft Windows XP SP1 or later.
2. Select a theme other than **Windows Classic** as the Windows theme. To verify or change the Windows theme, open the **Display** window in Control Panel. On the **Themes** tab, select a theme other than **Windows Classic** in the **Theme** list box.
3. On the **Tools** menu, click **Options**. On the **Forms/Reports** tab, select the **Use Windows Themed Controls on Forms** check box.

[Change the theme that is applied to form controls](#)

1. Open the **Display** window in Control Panel.
2. On the **Themes** tab, select a theme other than **Windows Classic** in the **Theme** list box.

[Remove theming of form controls](#)

1. On the **Tools** menu, click **Options**.
2. On the **Forms/Reports** tab, clear the **Use Windows Themed Controls on Forms** check box.

[For data access pages](#)

[Apply a theme](#)

[To a new data access page](#)

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Page** dialog box, click **Page Wizard** and follow the instructions

in the wizard.

4. On the last screen in the wizard, select the **Do you want to apply a theme to your page** check box, and then click **Finish**.
5. In the **Theme** dialog box, under **Choose a Theme**, click the theme you want.
6. Select the options you want.

[For an existing data access page](#)

1. In [Design view](#), open the data access page.
2. On the **Format** menu, click **Theme**.
3. In the **Choose a Theme** list, click the theme you want.
4. Select the options you want.

[Change the theme of a data access page](#)

1. In [Design view](#), open the data access page.
2. On the **Format** menu, click **Theme**.
3. In the **Choose a Theme** list, click the theme you want.
4. Select the options you want.

[Remove a theme](#)

[From a data access page](#)

1. In [Design view](#), open the data access page.
2. On the **Format** menu, click **Theme**.
3. In the **Choose a Theme** list, click **(No Theme)**.

[From all new data access pages](#)

1. In [Design view](#), open any data access page.
2. On the **Format** menu, click **Theme**.
3. In the **Choose a Theme** list, click **(No Theme)**.
4. Click **Set Default**.
5. Click **Yes**.

[Show All](#)

Show title and other informational text

[On a form or report](#)

1. Click the **Label** tool  on the [toolbox](#).
2. Click the section on the form or report where you want to place the label.
3. Click outside the label, then double-click it to display its property sheet.
4. Set other properties to customize the visual appearance of the label and its contents.

[On a data access page](#)

1. Open the data access page in [Design view](#).
2. To add a title or other text, click the placeholder text labeled **Click here and type title text**, and then type what you want.

Note Press ENTER to start a new paragraph.

3. Use the buttons on the **Formatting (Page)** toolbar to change the font, font size, and other characteristics of the text.

Note To display HTML-formatted text on a page, add a [bound span control](#) to the body or to a [section](#).

[Show All](#)

Show current date and time

1. Open the form or report in [Design view](#).
2. On the **Insert** menu, click **Date And Time**.
3. To include a date, select the **Include Date** check box, and then click a date format.
4. To include the time, select the **Include Time** check box, and then click a time format.

Note Microsoft Access adds two [text boxes](#), one to show the date and another to show the time, to the form or report and sets their **ControlSource** property to an appropriate expression. If there is a form or [report header](#), Access adds the text boxes to that section. Otherwise, it adds the text boxes to the [detail section](#). You can move, size, or position the text boxes and set their properties to customize their appearance.

[Show All](#)

Show page numbers

1. Open the form or report in [Design view](#).
2. On the **Insert** menu, click **Page Numbers**.
3. In the **Page Numbers** dialog box, select the format, position, and alignment for the page number. For alignment, the following options are available:

Left Page number shows at the left margin.

Center Page number is centered between the left and right margins.

Right Page number shows at the right margin.

Inside Odd page numbers print on the left, and even page numbers print on the right.

Outside Even page numbers print on the left and odd page numbers print on the right.

4. To show a number on the first page, select the **Show Number On First Page** check box.

Note Microsoft Access adds a [text box](#) to either the page header or the page footer based on your selection. The **ControlSource** property is set to an expression based on the format you chose. You can move, size, or position the text box, and set its properties to customize its appearance.

[Show All](#)

Set the size and position of a form or report

[Set the default size of a form or report](#)

1. Open a form in [Form view](#) or a report in Print Preview.
2. Size the form or report by dragging the borders of the window.
3. Click **Save**  on the toolbar.

Note If the **AutoResize** property is set to **Yes**, Microsoft Access automatically sizes the form or report to display complete records. To set the form or report window to the size you want, make sure the **AutoResize** property is set to **No**. The **AutoResize** property for a report is not available in Microsoft Access 2000 or earlier.

[Center a form or a report in the application window when it's first opened](#)

1. Open a form or report in [Design view](#).
2. Double-click the [form selector](#) or [report selector](#) to open the property sheet.
3. Set the **AutoCenter** property to **Yes**.

The next time you open the form or report, it will be centered in the application window.

Note The **AutoCenter** property for a report is not available in Microsoft Access 2000 or earlier.

[Size a form to fit its contents](#)

1. Open a form in [Form view](#).
2. On the **Window** menu, click **Size To Fit Form**. If the Form window is maximized, the command is not available.
3. Click **Save**  on the toolbar to save the size of the form.

Note If the form's **DefaultView** property is set to **Single Form**, and the

displayed record is smaller than the Form window, **Size To Fit Form** crops the window to the size of the record. If the record is larger than the Form window, **Size To Fit Form** expands the Form window to show as much of the record as possible. If the property is set to **Continuous Forms**, and only part of the bottom record is displayed, **Size To Fit Form** crops that record. If only one record is partially displayed, **Size To Fit Form** enlarges the window to show as much of the record as possible.

[Show All](#)

Customize a form or report window

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or [report selector](#) to display the property sheet.
3. Do one or more of the following:

[Change the title bar text](#)

1. In the Caption property box, type the title bar text.
2. To see the title bar text of a form, switch to Form view. To see the title bar text of a report, switch to preview mode.

[Tip](#)

If you don't want to have any title bar text, type a space in the **Caption** property box.

[Hide the Minimize and Maximize buttons](#)

- Set the **MinMaxButtons** property to **None**.

Note If the **BorderStyle** property for the form is set to **None** or **Dialog**, the form won't have **Minimize** and **Maximize** buttons even if the **MinMaxButtons** property is set to **Min Enabled**, **Max Enabled**, or **Both Enabled**.

[Disable the Close button](#)

- Set the **CloseButton** property to **No**.

Note When you set the **CloseButton** property to **No**, Microsoft Access disables the **Close** button and also disables the **Close** command on the **Control** menu. If you want to remove the **Close** button completely, set the **BorderStyle** property to **None**.

[Hide the control menu](#)

- Set the **ControlBox** property to **No**.

Note When you set the **ControlBox** property to **No**, the **Minimize**, **Maximize**, and **Close** buttons are also hidden. If the **BorderStyle** property is set to **None**, the form or report won't have a **Control** menu even if the **ControlBox** property is set to **Yes**.

[Hide the shortcut menu in a form](#)

- Set the **ShortcutMenu** property to **No**.

[Hide the navigation buttons in a form](#)

- Set the **NavigationButtons** property to **No**.

[Hide the record divider lines in a continuous form](#)

- Set the **DividingLines** property to **No**.

[Hide the section divider lines in a form](#)

- Set the **DividingLines** property to **No**.

[Hide the record selectors in a form](#)

- Set the **RecordSelectors** property to **No**.

[Show All](#)

Show or hide shortcut menu

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the form's property sheet.
3. Set the **ShortcutMenu** property box to **No**.

[Show All](#)

Prevent a data-entry form or page from showing existing records

[Prevent a form from showing existing records](#)

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the form's property sheet.
3. Do one of the following:

If you don't want the form to display existing records, set the **DataEntry** property to **Yes**.

If you want the form to open with all records showing, set the **DataEntry** property to **No**.

[Prevent a data access page from showing existing records](#)

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. Do one of the following:

If you don't want the page to display existing records, set the **DataEntry** property to **True**.

If you want the page to open with all records showing, set the **DataEntry** property to **False**.

[Show All](#)

Show or hide the field list

[In Design view](#)

Use the [field list](#) to add [controls](#), which are bound to the underlying fields or columns in a form, report, or [data access page](#).

1. Open a form, report, or data access page in [Design view](#).
2. Click **Field List**  on the toolbar.

[In PivotTable view or PivotChart view](#)

Use the [field list](#) to view the list of fields that are available from the underlying database object and to add fields to the different areas in the view.

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. Click **Field List**  on the **PivotTable** or **PivotChart** toolbar.

Notes

- In the field list, the fields that are not bold are available but are not yet displayed in the view. The fields that appear in bold are already displayed in the view.
- You can scroll and click the expand indicators (⊕ or ⊖ boxes) in the field list to show or hide additional fields.

[Show All](#)

Show or hide the grid

1. Open a form, report, or [data access page](#) in [Design view](#).
2. On the **View** menu, click **Grid**.

[Show All](#)

Change the distance between dots on a grid

You can change the distance between the dots in the grid for a form, report, or [data access page](#).

1. Open a form, report, or data access page in [Design view](#).
2. Display the property sheet by doing one of the following:

For a form or report, double-click the [form selector](#) or the [report selector](#).

For a data access page, click **Select Page** on the **Edit** menu.

3. Change the distance between the dots by doing one or both of the following:

For the horizontal dots, type a new value for the **GridX** property. Higher numbers indicate less distance between dots.

For the vertical dots, type a new value for the **GridY** property.

Note The grid is preset to 24 dots per inch horizontally and 24 dots per inch vertically. If you're using centimeters as your unit of measure, the grid is set to 10 x 10. You can change these settings to any integer value from 1 to 64. Dots on the grid are invisible if you choose a setting that is greater than 24 dots per inch or 9 dots per centimeter.

[Show All](#)

Open or close the toolbox

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click **Toolbox**  on the toolbar.

Lock or unlock a toolbox tool

When a toolbox tool is locked down, you don't have to click the button each time to perform a repeated action. For example, if you want to add several labels to a form, you can lock down the **Label** tool in the toolbox.

- To lock a tool, double-click the tool.
- To release the lock on a tool, press ESC.

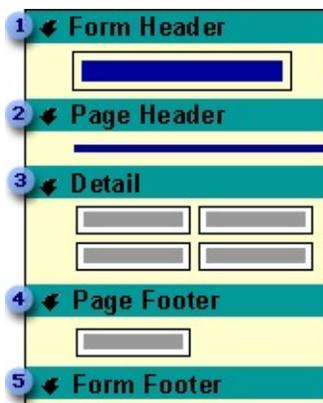
[Show All](#)

About sections of a form

The information in a form can be divided into sections. All forms have a [detail section](#), but a form can also include form header, [page header](#), [page footer](#), and form footer sections. Each section has a specific purpose and prints in a predictable order in the form.

In [Design view](#), sections are represented as bands, and each section that the form contains is represented once. In a printed form, the page header and footer might be repeated once for each page. You determine where information appears in every section by placing [controls](#), such as [labels](#) and [text boxes](#).

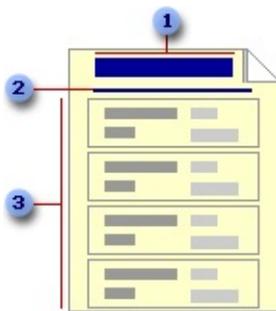
This illustration shows the Add Products form in Design view.



- 1 A form header displays information that remains the same for every record, such as a title for the form. A form header appears at the top of the screen in [Form view](#) and at the top of the first page when printed.
- 2 A page header displays information such as a title or column headings at the top of every printed page. Page headers appear only on printed forms.
- 3 A detail section displays records. You can either display one record on the screen or page, or display as many as will fit.
- 4 A page footer displays information such as the date or page number at the bottom of every printed page. Page footers appear only on printed forms.

5 A form footer displays information that remains the same for every record, such as [command buttons](#) or instructions for using the form. A form footer appears at the bottom of the screen in Form view or after the last detail section on the last page when printed.

This illustration shows the Add Products form as it will appear when printed.



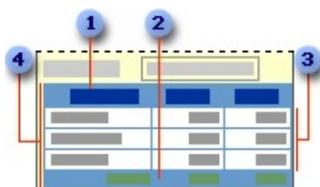
- 1 The form header prints at the top of the first page.
- 2 The page header prints at the top of every page, but appears only when printed or in Print Preview.
- 3 The detail section prints the data from the record source.

This illustration shows the Add Products form in Form view.



- 1 The form header displays a title.
- 2 The detail section displays data from the record source.

A [subform](#) can also have sections. For example, this subform has a form header and a form footer, in addition to the detail section.



- 1 Form header
- 2 Form footer
- 3 Detail section
- 4 Subform

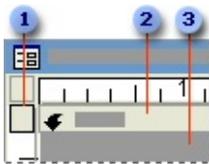
You can hide or resize a section, add a picture, or set the background color of a section. You can also set section properties to customize how the contents of a section will print.

[Show All](#)

Select a section

[Select a form or report section](#)

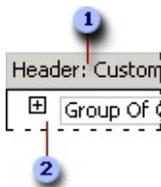
1. Open a form or report in [Design view](#).
2. Do one of the following:
 - Click the [section selector](#).
 - Click anywhere in the section bar.
 - Click anywhere in the background of the section.



- 1 Section selector
- 2 Section bar
- 3 Section background

[Select a data access page section](#)

1. Open a data access page in Design view.
2. Do one of the following:
 - Click the section bar.
 - Click anywhere in the background of the section.



- 1 Section bar
- 2 Section background

[Show All](#)

Add or remove a section

 [Add or remove a form or report section](#)

Form headers and footers appear at the top and bottom of a form in [Form view](#) and at the beginning and end of a printed form. [Report headers](#) and [footers](#) appear at the beginning and end of a printed report. [Page headers](#) and [footers](#) appear at the top and bottom of each printed page— they don't appear in Form view.

1. Open a form or report in [Design view](#).
2. Do one or more of the following:
 - To add or remove form header and footer, click **Form Header/Footer** on the **View** menu.
 - To add or remove report header and footer, click **Report Header/Footer** on the **View** menu.
 - To add or remove page header and footer on a form or report, click **Page Header/Footer** on the **View** menu.
 - To add or remove [group header](#) or [footer](#) on a report, do the following:
 1. Click **Sorting and Grouping**  on the toolbar.
 2. In the **Sorting and Grouping** dialog box, click a row in the **Field/Expression** column.
 3. To hide the group header section, in the lower portion of the dialog box, set the **Group Header** property to **No**.
 4. To hide the group footer section, set the **Group Footer** property to **No**.

Notes

- If you remove a header and footer, Microsoft Access deletes the [controls](#) in the header and footer.
- You can add a header and footer (except group header and footer) only as a pair. If you don't want both a header and a footer, you can set the **Visible** property of the section you don't want to **No**, or remove all controls from the section, and then size the section to zero height or set its **Height** property to **0**. You can also set the **DisplayWhen** property of a form header

and footer to specify when you want the header and footer to be displayed - always, in print only, or on the screen only.

[Add or remove a data access page section](#)

You can add or remove [sections](#) at the [group level](#) on a data access page that is bound to data. You can add an unbound section to a page that is not bound to any data.

1. Open a data access page in [Design view](#).
2. Do one or both of the following:

[Add or remove a group header, footer, caption, or record navigation section](#)

1. Click the arrow on the header [section bar](#) of the group for which you want to add or remove a section. If the group does not have a header, click the arrow on the group's footer section bar.
2. Do one or both of the following:
 - To add a section, click the section name.
 - To remove a section, clear the check box to the left of the section name.

[Tip](#)

You can remove a section and any controls in it by selecting the section and then pressing DELETE.

Notes

- You can't add a footer to the lowest group level on a page.
- You can't add bound controls to the [caption](#) and [record navigation sections](#).

[Add or remove an unbound section](#)

1. Open the data access page in Design view.
2. Do one of the following:

- To add an unbound section, click **Unbound Section** on the **Insert** menu.
- To remove the unbound section, click the [section bar](#) and press DELETE.

[Show All](#)

Show or hide a section

Hiding sections on a form or report is useful when you don't want to display information contained in a section.

1. Open a form or report in [Design view](#).
2. Double-click the [form selector](#) or the [report selector](#).
3. Set the **Visible** property to **Yes** or **No**.

Note You cannot hide the [page header](#) and [footer](#) sections on a form.

[Show All](#)

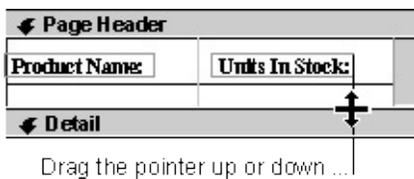
Resize a section

[Resize a form or report section](#)

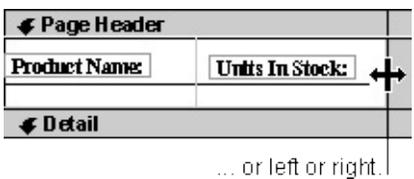
You can increase or decrease the height of form and report [sections](#) individually. However, when you change the width of one section, you change the width of the entire form or report.

1. Open the form or report in [Design view](#).
2. Do one or more of the following:

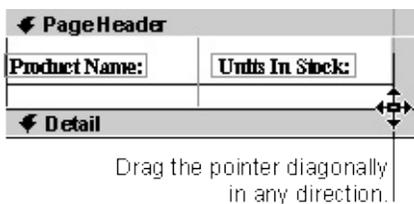
To change the height of a section, place the pointer on the bottom edge of the section and drag the pointer up or down.



To change the width of a section, place the pointer on the right edge of the section and drag the pointer left or right.



To change both height and width of a section, place the pointer in the lower-right corner of the section and drag it diagonally in any direction.



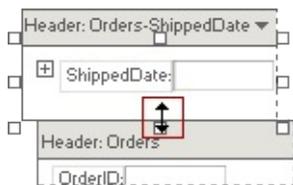
[Resize a data access page section](#)

You can increase or decrease the height of a data access page [section](#) individually. However, when you change the width of a section, Microsoft Access automatically resizes other sections that belong to the same [group level](#) to match the new width.

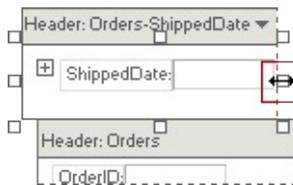
1. Open a data access page in [Design view](#).
2. Click the [section bar](#) of the section you want to resize.

Access displays selection handles around the perimeter of the section.

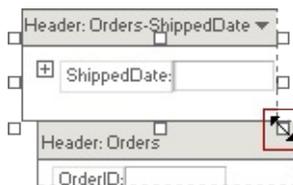
3. Do one or more of the following:
 - To change the height of a section, place the pointer on the bottom center selection handle and drag the pointer up or down.



- To change the width of a section, place the pointer on the selection handle that is in the middle of the right edge of the section, and drag the pointer left or right.



- To change both height and width of a section, place the pointer on the bottom right selection handle and drag it diagonally in any direction.



Notes

- You can drag any selection box to resize a section, but Access always adjusts a section from the bottom or the right edge. For example, if you

drag the selection box at the top of a section down to reduce the section's height, Access moves the bottom edge of the section up.

- You won't be able to drag the handles after the bottom or right edge of a section meets the edge of a control placed inside the section.

[Show All](#)

Keep the contents of a section together

The **KeepTogether** property is available for all form and report sections except [page headers](#) and [footers](#).

1. Open the form or report in [Design view](#).
2. Double-click the [section selector](#) to open the property sheet.
3. Set the **KeepTogether** property to **Yes**.

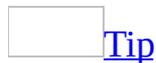
Notes

- If the section is longer than the print area on the page, the **KeepTogether** property setting is ignored.
- If you want to keep the data in a group together— for example, the [group header](#), detail records, and [group footer](#)— set the **KeepTogether** property for the group in the **Sorting and Grouping** dialog box.

[Show All](#)

Change the background color of a control or section

1. Open a form, report, or data access page in [Design view](#).
2. Click the [control](#) or [section](#) that you want to change.
3. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to **Fill/Back Color** .
4. Click a color in the palette.



If you want to use the same color for another control or section, you can just click **Fill/Back Color**  (you don't have to click the arrow next to the button and pick the color again from the palette). If you want to set different colors for several controls at once, click the arrow next to **Fill/Back Color** , click the horizontal bar at the top of the color palette, and drag it to another location. This way you don't need to click **Fill/Back Color**  each time you want to apply a color.

[Show All](#)

About subforms

A subform is a form that is inserted in another form. The primary form is called the [main form](#), and the form within the form is called the subform. A form/subform combination is often referred to as a hierarchical form, a master/detail form, or a parent/child form.

Subforms are especially effective when you want to show data from tables or [queries](#) with a [one-to-many relationship](#). For example, you could create a form with a subform to show data from a Categories table and a Products table. The data in the Categories table is the "one" side of the relationship. The data in the Products table is the "many" side of the relationship— each category can have more than one product.



- 1 The main form shows data from the "one" side of the relationship.
- 2 The subform shows data from the "many" side of the relationship.

The main form and subform in this type of form are linked so that the subform displays only records that are related to the current record in the main form. For example, when the main form displays the Beverages category, the subform displays only the products in the Beverages category.

[Displaying a subform in a form](#)

When you create a subform, you can design it to be displayed in Datasheet view, Form view, PivotTable view, or PivotChart view. You can also set the default view of the subform, and disable one or more views. You can switch the view of a subform when the main form is displayed in Form view.

Subforms are not displayed when a main form is opened in PivotTable or PivotChart view.

 [A subform displayed in Datasheet view](#)

Product Name:	Discontinued
Chai	<input type="checkbox"/>
Chang	<input type="checkbox"/>

 [A subform displayed as a simple continuous form in Form view](#)

Product Name: Chai
<input type="checkbox"/> Discontinued
<hr/>
Product Name: Chang
<input type="checkbox"/> Discontinued

 [A subform displayed as a continuous form in Form view](#)

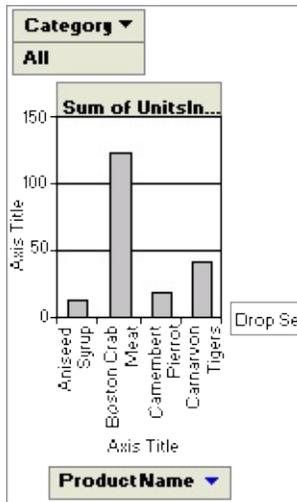
Product Name:	Qtr 1:	Qtr 2:
Aniseed Syrup		\$60.00
Chartreuse verte		\$283.50
Lakkalikööri		\$270.00
Totals:	\$0.00	\$613.50

You can make a form displayed in [Form view](#) look like a datasheet, but since it's displayed in Form view, it can display a form header and footer.

 [A subform displayed in PivotTable view](#)

Order Date		
7/8/1996		
	RequiredDate	GrandT
	8/5/1996	GrandT
	+ -	+ -
ShippedDate	OrderID	No To
7/15/1996	10250	
7/12/1996	10251	
Grand Total		

 [A subform displayed in PivotChart view](#)



[Synchronizing a form and a subform](#)

When you create a form and subform based on tables that have a one-to-many relationship, the main form shows the "one" side of the relationship and the subform shows the "many" side of the relationship. The main form is synchronized with the subform so that the subform shows only records related to the record in the main form.

If you use a wizard to create a subform, or if you drag a form, table, or query from the [Database window](#) to another form to create a subform, Microsoft Access automatically synchronizes the main form with the subform if one of the following conditions is met:

- You have defined [relationships](#) for the tables you select, or you have defined relationships for the tables underlying the queries you select.
- The main form is based on a table with a [primary key](#) and the subform is based on a table that contains a field with the same name as that primary key and with the same or a compatible data type and field size. For example, in a Microsoft Access database, if the primary key of the table underlying the main form is an AutoNumber field and its **FieldSize** property is set to **Long Integer**, the corresponding field in the table underlying the subform should be a Number field with its **FieldSize** property set to **Long Integer**. If you select a query or queries, the underlying tables for the query or queries must meet these conditions.

[Nesting levels](#)

A main form can have any number of subforms if you place each subform on the main form. You can also nest up to seven levels of subforms. This means you can have a subform within a main form, and you can have another subform within that subform, and so on. For example, you could have a main form that displays customers, a subform that displays orders, and another subform that displays order details. However, a form will not display subforms in PivotTable or PivotChart view.

[Entering data using a form and subform](#)

When you use a form with a subform to enter new records, Microsoft Access saves the current record in the main form when you enter data in the subform. This ensures that the records in the "many" table will have a record in the "one" table to relate to. It also automatically saves each record as you add it to the subform.

[Show All](#)

Work with subforms

[Change the layout of the subform](#)

How you change the layout of a [subform](#) depends on the view in which the subform is displayed.

[In Datasheet view](#)

1. Open the [main form](#) in Datasheet view.
2. Do one or more of the following in the subform:

To change the width of a column, drag the right edge of the [column selector](#). To automatically size the column to fit its data, double-click the right edge of the column selector.

To change the order of columns, click a column heading to select the column. Then click the column heading again, hold down the mouse button, and drag the column to a new location.

To change the height of all the rows, drag the bottom edge of the [row selector](#).

To hide a column, click anywhere in the column, and then click **Hide Columns** on the **Format** menu.

To show a hidden column, click **Unhide Columns** on the **Format** menu, and select the check box of the column you want to show.

To freeze a column, click anywhere in the column, and then click **Freeze Columns** on the **Format** menu.

[In Form view as a single or continuous form](#)

1. Open the [main form](#) in [Design view](#). The subform is displayed in Design view in the [subform control](#).
2. Right-click the subform control and click **Subform in New Window** to

open it in a separate window.

3. Make changes to the subform's design.
4. Save and close the subform window.
5. Switch the main form to Form view to see the changes.

[In PivotTable view](#)

1. Open the main form in [Form view](#). The subform is displayed in its default view.
2. On the **View** menu, point to **Subforms**, and then click **PivotTable**.
3. Add fields from the field list to the filter, row, column, or detail area. You can also move fields from one area to another.
4. Customize the data that is displayed in the view by filtering, grouping, or sorting data.

[In PivotChart view](#)

1. Open the main form in [Form view](#). The subform is displayed in its default view.
2. On the **View** menu, point to **Subforms**, and then click **PivotChart**.
3. Add fields from the field list to the filter, category, series, or data area. You can also move fields from one area to another.
4. Customize the data that is displayed in the view by filtering, grouping, or sorting data.

Notes

- If you change the width of a form displayed as a subform, you must also adjust the width of the [subform control](#) on the [main form](#).
- If the **AllowDesignChanges** property of the main form is set to **All Views**, you can change the design of the subform by using the property sheet in Form view. You can't, however, make changes to the subform control.

[Make a subform look like a datasheet that displays a header and a footer](#)

You can display a form header and footer on a subform and arrange the text boxes and other [controls](#) so that the subform looks like a datasheet when you open the form in [Form view](#).

1. Open the form that you want to use as the subform in [Design view](#).
2. If you haven't already added a form header and footer, on the **View** menu, click **Form Header/Footer**.
3. In the [detail section](#), arrange the text boxes and other controls that display data in a single row.
4. In the form header, arrange the labels for the text boxes and other controls. Make sure that you align the labels in the form header with their associated controls in the detail section.
5. In the form footer, if you want to total the records in the subform, create [calculated controls](#).
6. Size each section appropriately. For example, make the detail section the size of one row in a datasheet. If you don't have any controls in the form footer, set its **Height** property to **0** or set its **Visible** property to **No**.
7. Double-click the [form selector](#) to open the property sheet for the form.
8. In the **DefaultView** property box, click **Continuous Forms**. This setting displays records in a form one after another, similar to a datasheet.

Notes

- When you set **DefaultView** to **Datasheet**, Microsoft Access hides the header and footer when you display the form in Form view.
- The Quarterly Orders form in the Northwind sample database has a continuous subform that has a form header and footer and looks like a datasheet. To view this form, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Quarterly Orders form in Design view.



[Move between a main form and a subform in Form view](#)

Do one of the following:

- To switch from the [main form](#) to the [subform](#), click anywhere in the subform.
- To switch from the subform back to the main form, click a [control](#) on the main form (not the background of the main form).
- To move from the last field in the main form to the first field in the subform, press TAB.
- To move from the first field in the subform back to the last field in the main form, press CTRL+SHIFT+TAB.

- To move from the last field in the subform to the next field on the main form or to the first field in the next record or next page of the main form, press CTRL+TAB. If the **Cycle** property of the form is set to **Current Record** or **Current Page**, you must use the [navigation buttons](#) or other methods provided on the form to move to the next record.

[Show All](#)

Create a subform

If the [subform](#) will be linked to the [main form](#), make sure the underlying [record sources](#) are related before using this procedure.

You cannot add a subform to a form when it is displayed in PivotTable or PivotChart view.



[Create a form and subform at the same time](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, double-click **Form Wizard**.
4. In the first wizard dialog box, select a table or [query](#) from the list. For example, to create a Categories form that displays products for each category in a subform, select the Categories table (the "one" side of the [one-to-many relationship](#)).

Note It doesn't matter which table or query you choose first.

5. Double-click the fields you want to include from this table or query.
6. In the same wizard dialog box, select another table or query from the list. Using the same example, select the Products table (the "many" side of the one-to-many relationship).
7. Double-click the fields you want to include from this table or query.
8. When you click **Next**, if you set up the [relationships](#) correctly before starting the wizard, the wizard asks which table or query you want to view by. Using the same example, to create the Categories form, click **By Categories**.
9. In the same wizard dialog box, select the **Form With subform(s)** option.
10. Follow the directions in the remaining wizard dialog boxes. When you click **Finish**, Microsoft Access creates two forms, one for the [main form](#) and [subform control](#), and one for the subform.



[Create a subform and add it to another form](#)

1. Open the form that you want to add the subform to in [Design view](#).
2. Make sure the **Control Wizards** tool  in the [toolbox](#) is pressed in.
3. Click the **Subform/Subreport** tool  in the toolbox.
4. On the form, click where you want to place the subform.
5. Follow the directions in the wizard dialog boxes.
6. When you click **Finish**, Microsoft Access adds a [subform control](#) to your form. It also creates a separate form for the subform.

[Add a form to another form to create a form with a subform](#)

1. Open the form that you want to use as the main form in [Design view](#). This is usually a form based on a table or [query](#) on the "one" side of a [one-to-many relationship](#).
2. Make sure the **Control Wizards** tool  in the [toolbox](#) is pressed in.
3. Press F11 to switch to the [Database window](#).
4. Drag a form, table, or query from the Database window to the main form.

[Create a form with two subforms](#)

The main form has a [one-to-many relationship](#) with the first subform, and the first subform has a one-to-many relationship with the second subform, and the main form contains both the [subform controls](#).

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, double-click **Form Wizard** in the list.
4. In the first wizard dialog box, select the table or [query](#) for the main form. For example, suppose you want to create a Customers form that has two subforms— an Orders subform and an Order Details subform. Select the Customers table (the "one" side of the first one-to-many relationship).

Note It doesn't matter which table or query you choose first.

5. Double-click the fields you want to include from this table or query.
6. In the same wizard dialog box, select the table or query for the first subform. Using the same example, select the Orders table (the "many" side of the first one-to-many relationship), and then double-click the fields you want to include from this table or query.
7. In the same wizard dialog box, select the table or query for the second

subform. Using the same example, select the Order Details table (the "many" side of the second one-to-many relationship), and then double-click the fields you want to include from this table or query.

8. When you click **Next**, if you set up the [relationships](#) correctly before starting the wizard, the wizard asks which table or query you want to view by. Using the same example, to create the Customers form, click **By Customers**.
9. In the same wizard dialog box, select the **Form With subform(s)** option.
10. Follow the directions in the remaining wizard dialog boxes. When you click **Finish**, Microsoft Access creates a main form with two subform controls. It also creates two other forms, one for each subform.

Notes

- When you use a wizard to create a form with two subforms, Microsoft Access synchronizes the subforms by attaching an [event procedure](#) to the **OnCurrent** property of the first subform. You can copy and modify this event procedure if you want to synchronize more subforms on the same form. To see this event procedure, open the first subform in [Design view](#), double-click the [form selector](#), and then click the **Build** button  next to the **OnCurrent** property.
- The Customer Orders form in the Northwind sample database is a form with two subforms. To view this form, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Customer Orders form in Design view.

[Create a form with nested subforms](#)

The main form has a [one-to-many relationship](#) with the first subform, and the first subform has a one-to-many relationship with the second subform, and the second subform is embedded in the first subform.)

1. Create a form with a subform.

[How?](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.

3. In the **New Form** dialog box, double-click **Form Wizard**.
4. In the first wizard dialog box, select a table or [query](#) from the list. For example, to create a Categories form that displays products for each category in a subform, select the Categories table (the "one" side of the [one-to-many relationship](#)).

Note It doesn't matter which table or query you choose first.

5. Double-click the fields you want to include from this table or query.
6. In the same wizard dialog box, select another table or query from the list. Using the same example, select the Products table (the "many" side of the one-to-many relationship).
7. Double-click the fields you want to include from this table or query.
8. When you click **Next**, if you set up the [relationships](#) correctly before starting the wizard, the wizard asks which table or query you want to view by. Using the same example, to create the Categories form, click **By Categories**.
9. In the same wizard dialog box, select the **Form With subform(s)** option.
10. Follow the directions in the remaining wizard dialog boxes. When you click **Finish**, Microsoft Access creates two forms, one for the [main form](#) and [subform control](#), and one for the subform.

Note Select the **Tabular** (the equivalent of [Form view](#)) option when the wizard asks what layout you want for the subform.

2. In [Design view](#) of the main form, select the [subform control](#) and click again inside the control to select the form, just as you would if you opened the subform in Design view from the [Database window](#).
3. Do one of the following:
 - Repeat step 1 to create the second subform by using the wizard.
 - Drag a form, table, or query from the Database window to the subform. Microsoft Access creates another subform inside the first subform and creates a form for the subform.

[Show All](#)

Change the layout of a subform

How you change the layout of a [subform](#) depends on the view in which the subform is displayed.



[In Datasheet view](#)

1. Open the [main form](#) in Datasheet view.
2. Do one or more of the following in the subform:

To change the width of a column, drag the right edge of the [column selector](#). To automatically size the column to fit its data, double-click the right edge of the column selector.

To change the order of columns, click a column heading to select the column. Then click the column heading again, hold down the mouse button, and drag the column to a new location.

To change the height of all the rows, drag the bottom edge of the [row selector](#).

To hide a column, click anywhere in the column, and then click **Hide Columns** on the **Format** menu.

To show a hidden column, click **Unhide Columns** on the **Format** menu, and select the check box of the column you want to show.

To freeze a column, click anywhere in the column, and then click **Freeze Columns** on the **Format** menu.



[In Form view as a single or continuous form](#)

1. Open the [main form](#) in [Design view](#). The subform is displayed in Design view in the [subform control](#).
2. Right-click the subform control and click **Subform in New Window** to open it in a separate window.
3. Make changes to the subform's design.

4. Save and close the subform window.
5. Switch the main form to Form view to see the changes.

[In PivotTable view](#)

1. Open the main form in [Form view](#). The subform is displayed in its default view.
2. On the **View** menu, point to **Subforms**, and then click **PivotTable**.
3. Add fields from the field list to the filter, row, column, or detail area. You can also move fields from one area to another.
4. Customize the data that is displayed in the view by filtering, grouping, or sorting data.

[In PivotChart view](#)

1. Open the main form in [Form view](#). The subform is displayed in its default view.
2. On the **View** menu, point to **Subforms**, and then click **PivotChart**.
3. Add fields from the field list to the filter, category, series, or data area. You can also move fields from one area to another.
4. Customize the data that is displayed in the view by filtering, grouping, or sorting data.

Notes

- If you change the width of a form displayed as a subform, you must also adjust the width of the [subform control](#) on the [main form](#).
- If the **AllowDesignChanges** property of the main form is set to **All Views**, you can change the design of the subform by using the property sheet in Form view. You can't, however, make changes to the subform control.

[Show All](#)

Make a subform look like a datasheet that displays a header and footer

You can display a form header and footer on a subform and arrange the text boxes and other [controls](#) so that the subform looks like a datasheet when you open the form in [Form view](#).

1. Open the form that you want to use as the subform in [Design view](#).
2. If you haven't already added a form header and footer, on the **View** menu, click **Form Header/Footer**.
3. In the [detail section](#), arrange the text boxes and other controls that display data in a single row.
4. In the form header, arrange the labels for the text boxes and other controls. Make sure that you align the labels in the form header with their associated controls in the detail section.
5. In the form footer, if you want to total the records in the subform, create [calculated controls](#).
6. Size each section appropriately. For example, make the detail section the size of one row in a datasheet. If you don't have any controls in the form footer, set its **Height** property to **0** or set its **Visible** property to **No**.
7. Double-click the [form selector](#) to open the property sheet for the form.
8. In the **DefaultView** property box, click **Continuous Forms**. This setting displays records in a form one after another, similar to a datasheet.

Notes

- When you set **DefaultView** to **Datasheet**, Microsoft Access hides the header and footer when you display the form in Form view.
- The Quarterly Orders form in the Northwind sample database has a continuous subform that has a form header and footer and looks like a datasheet. To view this form, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Quarterly Orders form in Design view.

[Show All](#)

Move between a main form and a subform in Form view

Do one of the following:

- To switch from the [main form](#) to the [subform](#), click anywhere in the subform.
- To switch from the subform back to the main form, click a [control](#) on the main form (not the background of the main form).
- To move from the last field in the main form to the first field in the subform, press TAB.
- To move from the first field in the subform back to the last field in the main form, press CTRL+SHIFT+TAB.
- To move from the last field in the subform to the next field on the main form or to the first field in the next record or next page of the main form, press CTRL+TAB. If the **Cycle** property of the form is set to **Current Record** or **Current Page**, you must use the [navigation buttons](#) or other methods provided on the form to move to the next record.

[Show All](#)

Calculate a total or other aggregate values

In Design view

 [Calculate a total or average on a form or report](#)

1. Open a form or report in [Design view](#).
2. Click the **Text Box** tool  on the [toolbox](#).
3. Do one of the following:
 - To calculate a total or average for a group of records, add the text box to the [group header](#) or [footer](#).
 - To calculate a grand total or average for all records in a form or report, add the text box to the header or footer of the form or report.
4. Select the text box, and then click **Properties**  on the toolbar.
5. In the **ControlSource** property box, type an [expression](#) that uses the **Sum** function to calculate a total or the **Avg** function to calculate an average.

 [Tip](#)

In a [Microsoft Access database \(.mdb\)](#), for a form or report, you can click the **Build** button  to use the [Expression Builder](#) to create the expression.

 [Example](#)



Detail			
ProductName	Quantity	Price	=[Quantity]*[Price]
OrderID Footer 1			
Order Total:			=Sum([Quantity]*[Price])
PageFooter			
Report Footer 2			
Grand Total:			=Sum([Quantity]*[Price])

- 1 This control calculates the total sales for each order.
- 2 This control calculates the total sales for all orders in the report.

Notes

- The Employee Sales by Country report in the Northwind sample database has an example of group totals and a grand total. To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Employee Sales by Country report in Design view.
- When computing a total in a form or report in Design view with an [aggregate function](#) such as **Sum** or a [domain aggregate function](#) such as **DSum**, you can't use the name of a [calculated control](#) in the **Sum** function. You must repeat the [expression](#) in the calculated control. For example:

=Sum([Quantity] * [UnitPrice])

However, if you have a [calculated field](#) in an underlying [query](#), for example:

=ExtendedPrice: [Quantity] * [UnitPrice]

you can use the name of that field in the **Sum** function, as shown here:

=Sum([ExtendedPrice])

[Calculate a percentage value on a report](#)

1. Open a report in [Design view](#).
2. Click the **Text Box** tool  on the [toolbox](#).
3. Do one or both of the following:
 - If the report does not calculate group totals, click the [group footer](#) to add a text box to display the group totals.
 - If the report does not calculate a grand total, add a text box to the [report footer](#) to display the grand total.
4. Do one or both of the following:
 - To calculate what percentage of a group total or the grand total each item is, add a text box to the [detail section](#).
 - To calculate what percentage of the grand total a group of items is, add a text box to the [group header](#) or [footer](#).

Note If your report includes multiple [group levels](#), place the text box in the

header or footer of the level for which you want Microsoft Access to calculate a percentage.

5. Select the text box, and then click **Properties**  on the toolbar.
6. In the **ControlSource** property box, type an expression that divides the smaller total by the larger total of which the smaller total is a part.

Tip

In a [Microsoft Access database \(.mdb\)](#), you can click the **Build** button  to use the [Expression Builder](#) to create the expression.

Example



- 1 The percentage of the grand total is calculated by ...
- 2 ... dividing the value of the DailyTotal control in the ShippedDate footer ...
- 3 ... by the value of the GrandTotal control in the report footer.

7. Set the **Format** property of the text box to **Percent**.

Note The Employee Sales by Country report in the Northwind sample database has an example of percentages. To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Employee Sales by Country report in Design view.

[Calculate a total or other aggregate values on a data access page](#)

Calculate an aggregate value

[When the control you want to use to calculate the aggregate value is](#)

[already on the page](#)

1. Open the data access page in [Design view](#).
2. Select the text box or [bound span control](#) that contains the values you want to aggregate.
3. On the toolbar, click the arrow next to the **AutoSum** Σ button, and then select the aggregate function that you want to use.

The results you see depend on the type of page and the group level to which the selected control belongs.

If the bound control belongs to

Then

An ungrouped

page, or the outermost group

level on a [grouped page](#)

Microsoft Access creates a new group level, and adds a bound span control to the footer section of the new group level.

An inner group level on a grouped page

Access adds a bound span control to the footer section of the parent group level. If the footer section is not shown, the aggregate control is placed in the header section of the parent group level.

Note The list of aggregate functions that you see in the **AutoSum** list depends on the type of data you want to summarize. For numeric values, you can choose from **Sum**, **Average**, **Min**, **Max**, **Count**, **StDev**, and **Any**. For text, date, and time values, you can choose from **Min**, **Max**, **Count**, and **Any**.

[When the control you want to use to calculate the aggregate value is not on the page](#)

This procedure does not apply to ungrouped pages. The page will not show the detail values of the field that is being used to calculate the aggregate values. If you want to show both individual and aggregate values, add the bound control to the page, and follow the previous procedure.

1. Open the data access page in [Design view](#).
2. If the [field list](#) is not displayed, click **Field List** on the **View** menu.

3. In the field list, click the field that has the values you want to aggregate.
4. Drag the field to the header or footer section where you want to show the aggregate values. For example, if the page has two group levels, Orders and Order Details, you can drag a field from the Order Details table to the header or footer section of the Orders group level.

The **Layout Wizard** dialog box appears.

5. In the wizard dialog box, select the layout you want, and click **OK**.

Microsoft Access creates a bound span control in the target section. The control will use the **Sum** function for numeric values and the **Count** function for all other data types.

Calculate subtotals and grand totals

 [Calculate a total based on an aggregate value](#)

- Select the aggregate control, and then click **AutoSum**  on the toolbar.

Microsoft Access creates a [bound span control](#) in the footer section of the next higher [group level](#). If the aggregate control belonged to the highest group level, a new group level will be created. The new control will use the same [aggregate function](#) as the aggregate control on which it is based.

Notes

- You can set the properties of the aggregate control, and move, resize, or format the control to make it look the way you want.
- You can change the control's aggregate function by setting the control's **TotalType** property to the aggregate function you want. When you change the aggregate function, the default alias and caption of the control change to reflect the underlying aggregate function. To prevent the control name from changing, type a new alias for the control in its **ControlSource** property. User-specified aliases don't change when the aggregate function is changed.
- Changing the aggregate function of a control does not affect other aggregate controls, such as those that calculate subtotals or grand totals.

In PivotTable view

[Calculate a total on a datasheet or form](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Add a total field](#)

You can add [total fields](#) to the [detail area](#), but not to the [row](#), [column](#), or [filter area](#).

1. Open a datasheet or form in [PivotTable view](#).
2. Click the field that contains the values you want to summarize.
3. On the **PivotTable** toolbar, click **AutoCalc** , and then click the summary function you want.

[Add a calculated total field](#)

You can add a [detail](#) or [total field](#) that uses an [expression](#) to calculate data.

1. Open a datasheet or form in [PivotTable view](#).
2. On the **PivotTable** toolbar, click **Calculated Totals and Fields** , and then do one of the following:
 - To create a detail field, click **Create Calculated Detail Field**.
 - To create a total field, click **Create Calculated Total**.
3. In the **Properties** dialog box, type the expression you want on the **Calculation** tab, and then click **Change**.

[Tip](#)

To add field references, select a field that you want to use from the **Insert Reference To** box, and then click the **Insert Reference To** button.

Notes

- To edit the expression that is being used in a calculated field, select the field, edit the expression, and then click **Change** on the **Calculation** tab of the **Properties** dialog box. The field values will be automatically recalculated using the new expression.
- By default, total fields that summarize numeric data use the **Sum** summary function. Fields that summarize other types of data use the **Count** summary function.

[Show All](#)

Calculate a value

[In a form or report](#)

1. Open a form or report in [Design view](#).
2. Click the **Text Box** tool  in the [toolbox](#).
3. In the [detail section](#), click where you want to place the text box.
4. Do one of the following:

Place the insertion point in the text box and type an [expression](#) that calculates a total.

Select the text box, click **Properties**  on the toolbar, and type the expression in the **ControlSource** property box. In a [Microsoft Access database \(.mdb\)](#), to use the [Expression Builder](#) to create the expression, click the **Bu**  **ild** button next to the **ControlSource** property box.

Notes

- If you need more room to type an expression in the **ControlSource** property box, press SHIFT+F2 to open the **Zoom** box.
- If a form or report is based on a [query](#), you may want to put the expression in the query instead of in a [calculated control](#). It may improve performance, and if you are going to calculate totals for groups of records, it's easier to use the name of a [calculated field](#) in an [aggregate function](#).

[In a data access page](#)

If you want to do complex calculations using spreadsheet functions, use the Microsoft Office Spreadsheet Component. For all other calculations, you can use the [bound span control](#).

[Using the Spreadsheet Component](#)

1. Open the data access page in [Design view](#).
2. If they aren't already on the page, add the controls that will contain the data

you want to use in the calculation.

3. Click the **Office Spreadsheet** tool  in the [toolbox](#), and then click where you want the upper-left corner of the spreadsheet to be on the data access page.
4. In any cell in the spreadsheet, enter a formula that uses the HTML **Host** function to refer to the fields used in the calculation, and the specific operation you want to perform on the field's values expressed as a combination of operators, identifiers, and values.

Example To multiply the value in the Quantity field by the value in the UnitPrice field, type:

=Host().Quantity.value*Host().UnitPrice.value

Example To multiply the value in the UnitPrice field by a percentage rate stored in cell C1, type:

=\$C\$1*Host().UnitPrice.value

Notes

- To see the name of a spreadsheet, select the control, and then click **Properties**  on the toolbar. The property sheet title bar displays the spreadsheet name.
- You can use spreadsheet functions, which, in Microsoft Access, are available only in this control.
- Most likely, you'll want the calculated value to display in an area that looks like the other controls on the data access page. To do this you can modify the appearance of the spreadsheet so it looks like a text box.

[Using the bound span control](#)

1. Open the data access page in [Design view](#).
2. If the expression will reference field names, the fields must be in the page's [data definition](#). Either add the fields to the page as [bound span controls](#), or add the fields as bound text boxes.

[Tip](#)

The page will load faster in [Page view](#) and in Microsoft Internet Explorer if you use bound span controls.

3. Create another control by clicking the **Bound Span** tool  or the **Text Box** tool  in the [toolbox](#), and then click where you want the upper-left corner of the control to be on the page.
4. To display the property sheet, make sure that the control is selected, and then click **Properties**  on the toolbar.
5. In the **ControlSource** property box, type an alias and an appropriate [expression](#). To reference fields on the page, use the names of the fields in the expression. For example, if you want to compute the extended price for a record in the Order Details table, type:

ExtendedPrice: UnitPrice*Quantity

If you don't specify an alias, you must type an equal sign (=) before the expression. Microsoft Access will automatically supply an alias— for example, Expr1.

[Show All](#)

About calculating a total in a subform and displaying it on a form

You can calculate a subtotal in a [subform](#) and display it in the [main form](#).

In the following example, the main form and the subform are based on tables with a [one-to-many relationship](#). The main form shows records from the "one" side of the relationship; the subform shows records from the "many" side of the relationship.

Product	Unit Price	Quantity
	\$3.80	30
	\$12.80	40
	\$38.50	8

Subtotal: \$1,363.15

- 1 The main form, based on the Orders table, limits the records in the subform to those for the current order.
- 2 The subform, based on the Order Details table, calculates the order subtotal, but does not display the total.
- 3 The main form displays the order subtotal.

To calculate the order subtotal in the subform, you add a text box to the form footer of the subform and specify an [expression](#) in the **ControlSource** property box for the text box. The expression uses the **Sum** function. For example, the following expression calculates the total amount for all the products in the Orders form:

```
=Sum([Quantity] * [UnitPrice])
```

Detail	
ProductName	UnitPrice

Form Footer	
Order Subtotal	=Sum([Quantity]*[UnitPrice])

1

1 The form footer isn't displayed if the **DefaultView** property is set to **Datasheet**.

To display the order subtotal on the main form, create a text box on the main form that uses an expression to refer to the control on the subform that contains the calculated value. For example, you could specify the following expression in the **ControlSource** property box of the text box on the main form:

```
=[Orders Subform]![OrderSubtotal]
```

Notes

- If you want to use the value from a [calculated control](#) when computing a total, you can't use the name of the control in the **Sum** function. Instead, you can define the expression in the form's underlying [query](#) or [SQL statement](#) and use the name of the [calculated field](#) in the **Sum** function.
- The Orders form in the Northwind sample database has an example of a subform that calculates a subtotal that's displayed on the main form. To view this form, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Orders form in [Design view](#).

[Show All](#)

Validate or restrict data in a form

By validating or restricting data, you can give users immediate feedback about the data just entered and ensure that they get the correct information into a text box or other [control](#) on a form. You can create an [input mask](#) or a validation rule for a control that accepts data from users. To prevent a user from entering data, you can lock a control.

[Create an input mask for a control](#)

1. Open a form in [Design view](#).
2. Select a text box or a combo box, and then click **Properties**  on the toolbar to open the control's property sheet.
3. In the **InputMask** property box, do one of the following:

Type the [input mask](#) definition.

In a [Microsoft Access database \(.mdb\)](#), click the **Build** button  in the **InputMask** property box to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.

[Create a validation rule for a control](#)

1. Open a form in [Design view](#).
2. Select a [control](#), and then click **Properties**  on the toolbar to open the control's property sheet.
3. In the **ValidationRule** property box, do one of the following:

Type the validation rule.

In a [Microsoft Access database \(.mdb\)](#), click the **Build** button  to use the Expression Builder to create the validation rule.

4. In the **ValidationText** property box, type the error message that you want displayed if the text entered violates the validation rule.

Note In an Access database, if you set the **ValidationRule** property for the field in table Design view and also set the **ValidationRule** property for a control bound to that field, Access enforces both rules—the control validation rule doesn't override the field validation rule.

[Lock or disable a control](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Make sure that the [control](#) is selected, and then click **Properties**  on the **Form Design, Report Design, or Page Design** toolbar to open its property sheet.
3. In a form or report, do one of the following:
 - If you want to disable the control completely so that it's dimmed and can't receive the [focus](#), set the **Enabled** property to **No**.
 - If you want to make data in the control readable, but not allow users to change the data, set the **Locked** property to **Yes**. If you set the **Enabled** property to **No** and the **Locked** property to **Yes**, the control won't appear dimmed, but it won't be able to receive the focus.

In a data access page, to disable the control completely so that it's dimmed and can't receive the focus, set the **Disabled** property to **True**.

[Tip](#)

In a form or report, you can use the **Enabled** property together with the **AfterUpdate** property to disable a control until a user enters text or makes a selection in another control. For example, you might want to disable a command button until data has been entered in a text box. You can use the **AfterUpdate** property of the text box to call an [event procedure](#) or a macro to enable the command button.

Notes

- If you need more room to type in a property box, press SHIFT+F2 to open the **Zoom** box.
- Microsoft Access validates a field when you change data and try to leave the field, save the record, switch views, or close the form.

[Show All](#)

Disable a view for a form

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the form's property sheet.
3. Do one or more of the following:
 - To disable Form view, set the **AllowFormView** property to **No**.
 - To disable Datasheet view, set the **AllowDatasheetView** property to **No**.
 - To disable PivotTable view, set the **AllowPivotTableView** property to **No**.
 - To disable PivotChart view, set the **AllowPivotChartView** property to **No**.

Notes

- The views displayed in the **View** button list and on the **View** menu depend on the setting of these properties.
- You can use this property with the **DefaultView** property to control how users view a form.
- These properties are not available in Microsoft Access 2000 or earlier. To disable a view in a previous version of Access, set the **ViewsAllowed** property of a form.

[Show All](#)

Make a form or data access page read-only

You can make a form read-only by setting form properties. To make a data access page read-only, you have make each group in the page read-only by setting group level properties.



[Make a form read-only](#)

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the form's property sheet.
3. In the **AllowEdits** property box, click **No**.
4. In the **AllowDeletions** property box, click **No**.
5. In the **AllowAdditions** property box, click **No**.



[Make a group in a data access page read-only](#)

1. Open the data access page in [Design view](#).
2. Right-click the section bar of a section that belongs to the group you want to make read-only, and then click **Group Level Properties**.
3. In the **AllowAdditions** property box, click **False**.
4. In the **AllowDeletions** property box, click **False**.
5. In the **AllowEdits** property box, click **False**.

[Show All](#)

Specify whether a user can set form properties

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the property sheet for the form.
3. To allow the user to change property settings, set the **AllowDesignChanges** property to **All views**.

To prevent the user from changing property settings, set the **AllowDesignChanges** property to **Design View Only**.

[Show All](#)

Specify whether a user can add, edit, or delete a record

[In a form](#)

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to open the form's property sheet.
3. Do one or more of the following:

To prevent additions, set the **AllowAdditions** property to **No**.

To prevent edits, set the **AllowEdits** property to **No**.

To prevent deletions, set the **AllowDeletions** property to **No**.

[In a data access page](#)

1. Right-click the section bar of a section that belongs to the group for which you want to set properties, and then click **Group Level Properties**.
2. Do one or more of the following:

To prevent additions, set the **AllowAdditions** property to **False**.

To prevent edits, set the **AllowEdits** property to **False**.

To prevent deletions, set the **AllowDeletions** property to **False**.

Note The default setting of the properties in pages created using Access 2002 or later is **True**. In pages created using Access 2000, the properties are, by default, set to **False**. You can open a page created using Access 2000 in Access 2002 or later and set the properties to **True**.

[Show All](#)

Specify whether a user can move or resize a form or report

You can prevent a user from moving or resizing a form in Form and Datasheet views. Similarly, you can prevent a user from moving or resizing a report in print and preview modes.

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or [report selector](#) to open the property sheet.
3. Do one or both of the following:

To prevent a user from moving a form or report, set the **Movable** property box to **No**.

To prevent a user from resizing a form or report, set the **BorderStyle** property to **None**, **Thin**, or **Dialog**.

Notes

- If the **Movable** property is set to **No**, the **Min**, **Max**, and **Close** buttons on the form or report window, if present, will continue to work. The user will be able to size the window, unless the **BorderStyle** property has been set to a value other than **Sizable**. The user will also be able to tile the window.
- The **Movable** property is not available for reports in Microsoft Access 2000 or earlier.

About PivotTable forms

Note A PivotTable form is different from a form that is opened in PivotTable view. You cannot create PivotTable forms in Microsoft Access 2002 or later. Instead, you can open an existing form in PivotTable view and build the layout of the view, or use the PivotTable wizard to help you design the PivotTable view for a new form.

You can dynamically change the layout of a PivotTable form to analyze data in different ways. You can rearrange row headings, column headings, and page fields until you achieve the desired layout. Each time you change the layout, the PivotTable form immediately recalculates the data based on the new arrangement. In addition, as source data changes, you can update the PivotTable form.

A PivotTable form performs the calculations you choose, such as sums (the default for numeric fields) and counts (the default for text fields), based on how the data is arranged in the PivotTable form. For example, a PivotTable form can display a field's values horizontally or vertically, and then calculate the total of the row or column.

A PivotTable form can also use a field's values as row or column headings, calculating individual amounts at the intersection of each row and column heading, and then calculating subtotals and grand totals. For example, to analyze sales by product category for each employee, you can list employee names as column headings across the top of the PivotTable form, product categories as row headings down the side of the PivotTable form, and the sales amount calculated by product category for each employee at the intersection of each row and column.

You can limit the records that the PivotTable form uses by designating a field as a page field—which becomes a drop-down list that works as a filter above the PivotTable form. The value you select from the list determines which records are displayed in the PivotTable form. For example, you could use the City field from the Employees table as a page field to view the sales performance by product category of just the sales representatives in London.

Each time you open a PivotTable form, the PivotTable form shows you the data that was current the last time you activated the PivotTable form. To display the most current data in the PivotTable form, you need to update the data.

[Show All](#)

Edit a PivotTable form

Note A PivotTable form is different from a form that is opened in PivotTable view. You cannot create PivotTable forms in Microsoft Access 2002 or later. Instead, you can open an existing form in PivotTable view and build the layout of the view, or use the PivotTable wizard to help you design the PivotTable view for a new form.

Because a PivotTable form is an embedded Microsoft Excel object, you modify it by going to the Microsoft Access form where the PivotTable object is embedded, and then starting and using Microsoft Excel to edit the data.

1. Open the PivotTable form in [Form view](#).
2. Click the **Edit PivotTable Object** button.

Note If you prefer to use [in-place editing](#) instead of editing the PivotTable form in a separate Microsoft Excel window, point to **Worksheet Object** on the **Edit** menu in Microsoft Access, and then click **Edit**.

3. On the **Data** menu in Microsoft Excel, click **Refresh Data**.
4. Make the changes you want to the PivotTable form.

Note For help on specific ways you can modify or otherwise work with the PivotTable form, use Microsoft Excel Help.

5. When you're done editing the PivotTable form, click **Exit** on the **File** menu in Microsoft Excel.

Notes

- If you don't find the answer you need in the Microsoft Excel Help topics, you can use Microsoft Access Help when the PivotTable form is open in Form view in Microsoft Access. If you're using in-place editing, you must first click any place on the Access form outside the PivotTable object to get Microsoft Access Help.
- The source data that the PivotTable form uses isn't automatically saved with the PivotTable form.

[Show All](#)

Refresh the data in a PivotTable form

Note A PivotTable form is different from a form that is opened in PivotTable view. You cannot create PivotTable forms in Microsoft Access 2002 or later. Instead, you can open an existing form in PivotTable view and build the layout of the view, or use the PivotTable wizard to help you design the PivotTable view for a new form.

1. Open the PivotTable form in [Form view](#).
2. Click the **Edit PivotTable Object** button.
3. On the **Data** menu in Microsoft Excel, click **Refresh Data**.

[Show All](#)

Save a copy of a PivotTable form's data

Note A PivotTable form is different from a form that is opened in PivotTable view. You cannot create PivotTable forms in Microsoft Access 2002 or later. Instead, you can open an existing form in PivotTable view and build the layout of the view, or use the PivotTable wizard to help you design the PivotTable view for a new form.

If you know that you'll need to edit a PivotTable form later on when you won't have access to the data, save a copy of the data with the PivotTable form.

1. Open the PivotTable form in [Form view](#).
2. Click the **Edit PivotTable Object** button.
3. On the **Data** menu in Microsoft Excel, click **Refresh Data**.
4. On the **Data** menu, click **PivotTable and PivotChart Report** to restart the PivotTable Wizard.
5. Click **Options**.
6. Under **Data Source Options**, make sure **Save Data With Table Layout** is selected.
7. Click **OK**.
8. Click **Finish**.
9. Continue working with the PivotTable form, or close it and return to the form in Microsoft Access by clicking **Close & Return To Pivot Form** on the Microsoft Excel **File** menu.

Notes

- Saving the data can significantly increase the size of the form, so make sure that you have enough disk space available.
- The changes you make to data saved with a PivotTable form apply only to that PivotTable form; they don't affect data in the underlying table.

[Show All](#)

Create, customize, and delete a switchboard form (MDB)

You create, customize, and delete a switchboard by using the Switchboard Manager.

 [Create a switchboard](#)

1. Open the database.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Switchboard Manager**.
3. If Microsoft Access asks if you'd like to create a switchboard, click **Yes**.
4. Click **New**.
5. Type the name of the new switchboard, and then click **OK**.

Microsoft Access adds the switchboard to the **Switchboard Pages** box.

6. Click the new switchboard, and then click **Edit**.
7. Click **New**.
8. Type the text for the first switchboard item in the **Text** box, and then click a command in the **Command** box. For example, type **View Recording Artists**, and then click **Open Form In Edit Mode** in the **Command** box.

Note To create a switchboard that branches to other switchboards, click the **Go To Switchboard** command in the **Command** box, and then specify the switchboard you want to go to.

9. Depending on which command you click, Microsoft Access displays another box below the **Command** box. Click an item in this box, if necessary. For example, if you clicked **Open Form In Edit Mode** in step 8, click the name of the form you want to open in the **Form** box, such as **Recording Artists**, and then click **OK**.
10. Repeat steps 7 through 9 until you've added all the items to the switchboard.

Note To edit or delete an item, click the item in the **Items On This Switchboard** box, and then click **Edit** or **Delete**. If you want to rearrange

items, click the item in the box, and then click **Move Up** or **Move Down**.

11. When you've finished creating the switchboard, click **Close**.

Notes

- To make a switchboard open when you open the database, click the switchboard name in the **Switchboard Manager** dialog box, and then click **Make Default**.
- When you create a switchboard with the Switchboard Manager, Microsoft Access creates a Switchboard Items table that describes what the buttons on the form display and do. If you make changes to the Switchboard form later in form [Design view](#), the application may no longer work.

[Customize a switchboard](#)

1. Open the database.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Switchboard Manager**.
3. In the **Switchboard Manager** dialog box, select the switchboard you want to edit, and then click **Edit**.
4. Click the item on the switchboard you want to change, and then do one of the following:

To change the text of the item, the command performed by the item, or the object that's opened or run when you click the item, click **Edit**.

To add an item, click **New**, type the text for the item, and then click a command in the **Command** box. Depending on which command you click, Microsoft Access displays another box below the **Command** box. Click an item in this box, if necessary. For example, if you click one of the form commands in the **Command** box, click the name of the form you want to open.

To delete an item, click **Delete**.

To move an item, click **Move Up** or **Move Down**.

5. Click **Close**.

[Delete a switchboard](#)

1. Open the database.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Switchboard Manager**.
3. In the **Switchboard Manager** dialog box, select the switchboard you want to delete, and then click **Delete**.
4. Click **Close**.

Note If you have not yet created the database for which you want to add a switchboard, you might want to use a database wizard. The wizard automatically creates a switchboard that helps you to navigate around the database.

Specify the default switchboard

1. Open the database.
2. Do one of the following to open the **Switchboard Manager** dialog box:

If you created the switchboard by using the Database Wizard, click **Change Switchboard Items** in the main switchboard.

If you created the switchboard by using the Switchboard Manager, point to **Database Utilities** on the **Tools** menu, and then click **Switchboard Manager**.

3. Select the switchboard that you want to use as the default switchboard.
4. Click **Make Default**.

The next time you open your database, Microsoft Access will automatically display the switchboard you specified.

[Show All](#)

About pop-up forms and reports

A pop-up form or report stays on top of other open [database objects](#), even when another object is active. You can create a pop-up form or report to display information to a user or to prompt a user for data.

A pop-up form or report can be either modeless or [modal](#).

Modeless pop-up forms and reports

When a modeless pop-up form or report is open, you can access other objects and menu commands. For example, on a Suppliers form, you could add a command button that displays a Product List pop-up form. The pop-up form displays the products available from the current supplier in the Suppliers form.



- 1 Click this button to display the modeless pop-up form.
- 2 The form stays on top of other windows, but you can move the focus to another window without closing the form.

Modal pop-up forms and reports: custom dialog boxes

When a modal pop-up form or report is open in a view other than Layout Preview, you can't access any other objects or menu commands unless you close the form or report. A modal pop-up form or report is called a custom dialog box. A custom dialog box usually contains one or more command buttons that run [macros](#) or [event procedures](#) that use the information entered in the dialog box to perform other actions.

You can create a custom dialog box that accepts criteria for generating a query or report. Or, you could create a custom dialog box that asks what report you want to print.



1 To move the focus to another window, you must close the window by clicking on either of these buttons.

A custom dialog box has a thick border and can include only a title bar and a **Control** menu. A modal form or report can't be maximized, minimized, or resized.

[Show All](#)

Create a pop-up form or report

Before you do the following procedure, create the form or report you want.

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or [report selector](#) to open the [property sheet](#).
3. In the **Pop Up** property box, click **Yes**.
4. Do one of the following:

[For a modeless pop-up form or report](#)

1. If you want to be able to size the form or report, in the **BorderStyle** property box, click **Thin**.
2. Create a macro or an event procedure to open the pop-up form or report.

Note The Suppliers form in Northwind sample application has an example of an event procedure that displays a modeless pop-up form. Open the Northwind database in the Samples subfolder of your Office folder, and then open the Suppliers form in Design view. Click the **Build** button next to the **OnClick** property box of the Review Products button.

[For a modal pop-up form or report](#)

1. In the **Modal** property box, click **Yes**.
2. In the **BorderStyle** property box, click **Dialog**.
3. Create a macro or an event procedure to open the pop-up form or report.
5. Attach the macro or an event procedure to a form or report by specifying the macro name or event procedure as the setting for the appropriate event property. For example, type the name of the macro or event procedure in the **OnClick** property box of a command button.

To position a pop-up form or report on the screen, open it in Design view,

maximize the Microsoft Access window, position the window where you want it, and then save the form or report. The pop-up window will appear in this location when the form or report is opened (assuming the **AutoCenter** property is set to **No** and you haven't specified its position in the event procedure that opens the pop-up object).

Note If you want to use a form as a normal form as well as a dialog box, you can use the Dialog setting of the OpenForm action to open the form temporarily as a dialog box instead of using this procedure.

[Show All](#)

Create a dialog box

You can create a dialog box by creating a form and setting certain properties to make the form function like a dialog box.

1. Create a form and add the controls you want on the dialog box to the form.
2. Open the form in [Design view](#).
3. Double-click the [form selector](#) to open the [property sheet](#).
4. In the **Pop Up** property box, click **Yes**.
5. In the **Modal** property box, click **Yes**.
6. In the **BorderStyle** property box, click **Dialog**.
7. In the **RecordSelectors** property box, click **No**.
8. In the **NavigationButtons** property box, click **No**.
9. In the **DividingLines** property box, click **No**.
10. Create a macro or an event procedure to open the dialog box.
11. Attach the macro or an event procedure to the form by specifying the macro name or event procedure as the setting for the appropriate event property. For example, type the name of the macro or event procedure in the **OnClick** property box of a command button.

To position a dialog box on the screen, open it in Design view, maximize the Microsoft Access window, position the window where you want it, and then save the form. The dialog box will appear in this location when it is opened (assuming the **AutoCenter** property is set to **No** and you haven't specified its position in the event procedure that opens the dialog box).

[Show All](#)

Property settings for pop-up forms and reports

The following table lists properties you can use to create or customize pop-up forms and reports. For more information on any of these properties, click the property in the property sheet and press F1.

Property	Determines
PopUp	Whether a form or report opens as a pop-up. A pop-up form or report always remains on top of other Microsoft Access windows.
Modal	Whether a form or report opens as a modeless object (you can switch to other windows) or a modal object (the form or report retains the focus until it's closed).
BorderStyle	The type of border and border elements (title bar, Close button <input type="checkbox"/> , Control menu, Maximize and Minimize buttons) to use for the form or report. It also determines whether the form or report is sizable.
ControlBox	Whether a form has a Control menu in Form view or a report has a Control menu in Print Preview.
MinMaxButtons	Whether a form has Minimize and Maximize buttons in Form view or a report has Minimize and Maximize buttons in Print Preview. If you set the BorderStyle property to Dialog , Microsoft Access automatically removes the Minimize and Maximize buttons.
CloseButton	Whether the Close button is disabled.
ScrollBars	Whether a form has scroll bars.
NavigationButtons	Whether a form has navigation buttons.
RecordSelectors	Whether a form has record selectors .
ShortcutMenu	Whether a shortcut menu is displayed when you right-click a form.
AutoCenter	Whether a form or report is centered automatically in the application window when it is opened.

AutoResize	Whether a form or report is resized automatically to display all of its contents.
AllowEdits	Whether a user can make any editing changes to saved records in a form.
AllowDeletions	Whether a user can delete records using a form.
AllowAdditions	Whether a user can add records using a form.
DataEntry	Whether a form shows existing records when it opens.
DefaultView	Whether a form is displayed as a single form (one record), as a continuous form (multiple records), in Datasheet view, in PivotTable view, or PivotChart view.
AllowFormView	Whether you can switch to Form view of a form.
AllowDatasheetView	Whether you can switch to Datasheet view of a form.
AllowPivotTableView	Whether you can switch to PivotTable view of a form.
AllowPivotChartView	Whether you can switch to PivotChart view of a form.

[Show All](#)

Specify a border and title for a pop-up form or report

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or [report selector](#) to open the property sheet.
3. In the **Caption** property box, type the title of the form or report.
4. Do one of the following:

For a modeless pop-up form or report, set the **BorderStyle** property to **Thin**.

For a [modal](#) pop-up form or report, set the **BorderStyle** property to **Dialog**.

[Show All](#)

Temporarily open a form as a dialog box

1. Create a [macro](#) or [event procedure](#) that uses an OpenForm action to open the form.
2. Set the Window Mode argument of the OpenForm action to Dialog (acDIALOG for an event procedure).

When you open a form that has its Window Mode argument set to Dialog, Microsoft Access sets the form's **PopUp** and **Modal** properties to **Yes** until the form is closed.

Note Setting the Window Mode argument to Dialog disables user-defined key combinations and some other Microsoft Access function keys, and suspends the macro or event procedure until you close or hide the form. You can hide a form by setting its **Visible** property to **No** using the SetValue action.

[Show All](#)

About reports

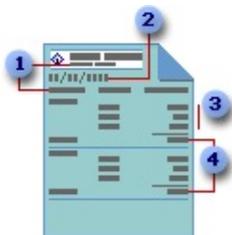
A report is an effective way to present your data in a printed format. Because you have control over the size and appearance of everything on a report, you can display the information the way you want to see it.



- 1 Create mailing labels.
- 2 Show totals in a chart.
- 3 Group records into categories.
- 4 Calculate totals.

Most reports are bound to one or more table and [query](#) in the database. A report's record source refers to the fields in the underlying tables and queries. A report need not contain all the fields from each of the tables or queries that it is based on.

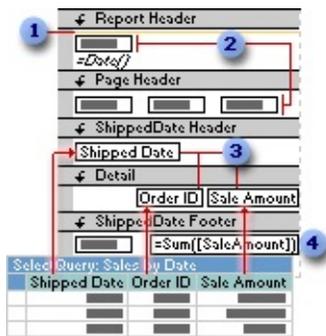
A bound report gets its data from its underlying record source. Other information on the form, such as the title, date, and page number, is stored in the report's design.



- 1 The report title and column headings are stored in the report's design.

- 2 The date comes from an expression stored in the report's design.
- 3 Data comes from fields in the underlying table, query, or SQL statement.
- 4 Totals come from expressions, which are stored in the report's design.

You create the link between a report and its record source by using graphical objects called [controls](#). Controls can be text boxes that display names and numbers, labels that display titles, or decorative lines that graphically organize the data and make the report more attractive.



- 1 Lines are decorative.
- 2 Labels display descriptive text.
- 3 Text boxes display data from the Sales by Date query.
- 4 A text box uses an expression to calculate a total.

[Creating a report](#)

You can create different types of reports quickly by using wizards. Use the Label Wizard to create mailing labels, the Chart Wizard to create charts, or the Report Wizard to create a standard report. The wizard asks you questions and creates a report based on your answers. You can then customize the report the way you want it in [Design view](#).

[Customizing a report](#)

You can customize a report in the following ways:

Record source Change the tables and queries that a report is based on.

Sorting and grouping data You can sort data in ascending or descending order. You can also group records on one or more fields, and display subtotals and grand totals on a report.

Report window You can add or remove **Maximize** and **Minimize** buttons, change the title bar text, and other Report window elements.

Sections You can add, remove, hide, or resize the header, footer, and details sections of a report. You can also set section properties to control the appearance and printing of a report.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a report.

[Show All](#)

Troubleshoot reports

[Printing and previewing](#)

[My report prints a blank page at the end of a printed report.](#)

Make sure the **Height** property is set to **zero** for the [report footer](#).

[Every other page in my report is blank.](#)

Make sure that the total width of the form or report plus the widths of the left and right margins don't exceed the paper size specified in the **Page Setup** dialog box (**File** menu). You can use the following formula as a guide:

report width + left margin + right margin <= paper size

You can adjust the size of the report in one or both of the following ways:

- Reduce the width of the report.
- Reduce the margins or change the page orientation.

[There is too much blank space in my report.](#)

You can reduce the blank space by doing one of the following:

- Reduce the width of the report.
- Move controls closer together.
- Reduce the size of a control to fit its contents.

[Layout Preview displays the wrong data.](#)

Layout Preview displays all the sections of a report and a few detail records. It sorts and groups the data it displays, but ignores any parameters, criteria, or joins

in the underlying [query](#). For example, the detail records you see in a group may not be the records that actually belong in that group.

To verify the accuracy of the data in a report, use Print Preview.

[My report prompts me for unexpected parameters.](#)

Make sure that all field names used in the following areas of the report's design are spelled correctly and match field names in the underlying table or [query](#):

- **ControlSource** property of any [bound controls](#)
- Field/Expression column in the **Sorting And Grouping** box
- Expressions in controls or in the **Sorting And Grouping** box

Also, make sure that any [aggregate functions](#) reference field names in the underlying table or query, not control names.

[Working with subreports](#)

[The column headings in my subreport don't print.](#)

Microsoft Access doesn't print the [page header](#) and [footer](#) in a subreport, so if you put the labels for the column headings in the subreport's page header, they won't show up when you print the report.

- If a [subreport](#) will always fit on one printed page, you can put the labels for the column headings in the subreport's [report header](#).
- If a subreport might span several printed pages, put the labels for the column headings in the subreport's [group header](#), and set the **RepeatSection** property of the group header to **Yes**.

[The data in my report and subreport is not synchronized. All records show up in the subreport.](#)

Link your main report and subreport using the **LinkChildFields** and **LinkMasterFields** properties.

1. Open the main report in [Design view](#).
2. Make sure the [subreport control](#) is selected, and then click **Properties**  on the toolbar to display the property sheet for the subreport control.
3. Set the linking fields. To enter more than one linking field or control, separate the field or control names with a semicolon. If you enter more than one field or control, you must enter the names in the same order in the **LinkChildFields** and **LinkMasterFields** properties.
 - In the **LinkChildFields** property box, enter the names of the linking fields in the subreport.
 - In the **LinkMasterFields** property box, enter the names of the linking fields or controls in the main report.

Note You can't use the name of a control in the **LinkChildFields** property box. The **LinkMasterFields** property can be set to the name of either a field or a control.

4. If you're unsure what the linking fields should be, click the **Build** button  next to either property box to open the Subform/Subreport Field Linker.

If you've already set the **LinkChildFields** and **LinkMasterFields** properties and the subreport still doesn't work, make sure that:

- You aren't using fields that have been renamed or deleted.
- You've spelled the field names correctly.
- You're using the actual field names, not the control names specified in the **Name** property.

Notes

- The linking fields don't have to appear in the main report or the subreport, but they must be included in the underlying [record source](#). If you use the Report Wizard to create the subreport, Microsoft Access automatically

includes the linking fields in the underlying record source even if you don't select them in the wizard.

- The linking fields must contain the same kind of data and have the same or a compatible [data type](#) or field size. For example, an AutoNumber field is compatible with a Number field whose **FieldSize** property is set to **Long Integer**.

[My subreport prints only one column when it should print more.](#)

A subreport might print only one column if:

- In the subreport, the **Column Layout** property is set to **Down, Then Across** on the **Columns** tab in the **Page Setup** dialog box, and
- In the main report, the **CanGrow** property is set to **Yes** for the subreport control.

You can redesign the report in one of the following ways:

- In the subreport, change **Column Layout** to **Across, Then Down** and, on the main report, leave the **CanGrow** property set to **Yes** for the subreport control. The subreport control will display multiple columns and expand as needed.
- In the subreport, leave **Column Layout** set to **Down, Then Across**. On the main report, size the subreport control to fit the maximum number of records, and set its **CanGrow** property to **No**. Because the subreport control will always be the same size, if the number of records is greater than the maximum number you anticipated, records that don't fit in the subreport control will be truncated.
- Or, instead of using a subreport, you could put the controls that print the data in the main report in a [group header](#) or the [page header](#), and put the controls that print the data in the subreport in the [detail section](#).

[Sorting and grouping](#)

[The records in the groups aren't sorted](#)

You must specify a sort order for the records in a group in the **Sorting And Grouping** box in report Design view. For example, if you want to sort the orders that were shipped each day by Order ID, group on Shipped Date and sort on Order ID.

[The records aren't sorted in the way I expect for my language.](#)

In a [Microsoft Access database](#) (.mdb), set the sort order for the database to the correct language in the **Options** dialog box.

[How?](#)

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab.
3. To specify the default language sort order for a new database, click the language that you want to use in the **New database sort order** box.

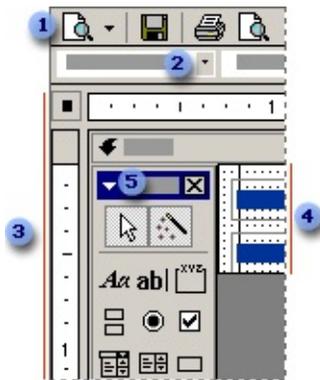
Note To reset the sort order for an existing database to the language that you selected in the **General** tab, compact that database.

[Show All](#)

About views of a report

[Design view](#)

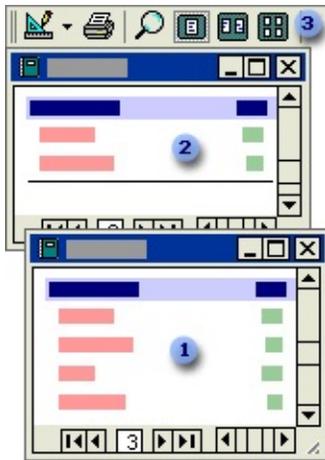
To create a report, you work in [Design view](#). Viewing a report in Design view is like sitting at a workbench surrounded by useful tools.



- 1 Click the **View** button to switch to Design view or to another view.
- 2 Use the **Formatting** toolbar to change the fonts or font sizes, align text, change border or line widths, or apply colors or special effects.
- 3 Align [controls](#) with the rulers.
- 4 Place controls in the [detail section](#), or one of the other report sections.
- 5 Use the toolbox to add controls, such as labels and text boxes.

[Print Preview and Layout Preview](#)

After you've created a report in Design view, you can preview it in Print Preview or Layout Preview.



- 1 In Print Preview, you can see how the report will look when printed.
- 2 In Layout Preview, you can preview the layout of the report. In this view, the report shows only a few sample records.
- 3 Use the Print Preview toolbar buttons to preview the report at different magnifications.

[Show All](#)

Create a report

There are three ways to create a report.

Based on a single table or query by using AutoReport. AutoReport creates a report that displays all fields and records in the underlying table or [query](#).

How?

1. In the [Database window](#), click **Reports**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click one of the following wizards:

AutoReport: Columnar Each field appears on a separate line with a label to its left.

AutoReport: Tabular The fields in each record appear on one line, and the labels print once at the top of each page.

4. Click the table or query that contains the data you want to base your report on.
5. Click **OK**.

Microsoft Access applies the last autoformat you used to the report. If you haven't created a report with a wizard before or haven't used the **AutoFormat** command on the **Format** menu, it uses the Standard autoformat.

Tip

You can also create a single-column report based on the open table or query or on the table or query selected in the Database window. Click **AutoReport** on the **Insert** menu, or click the arrow next to the **New Object** button on the toolbar, and then click **AutoReport**. Reports created with this method don't have a [report header](#) and [footer](#) or a [page header](#) and [footer](#).

Based on one or more tables or queries with a wizard. The wizard asks you

detailed questions about the record sources, fields, layout, and format you want and creates a report based on your answers.

[How?](#)

1. In the [Database window](#), click **Reports**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click the wizard that you want to use. A description of the wizard appears in the left side of the dialog box.
4. Click the table or [query](#) that contains the data you want to base your report on.

Note You don't need to do this step if you clicked **Report Wizard** in step 3 — you can specify the record source for the report in the wizard.

5. Click **OK**.
6. Follow the instructions in the wizard.

If the resulting report doesn't look the way you want, you can change it in [Design view](#).

Notes

- If you want to include fields from multiple tables and queries in your report, don't click **Next** or **Finish** after selecting the fields from the first table or query in the Report Wizard. Repeat the steps to select a table or query, and pick fields that you want to include in the report, until you have selected all the required fields.
- If you click one of the **AutoReport** options, Microsoft Access uses the [autoformat](#) you last specified, either in the Report Wizard or using the **AutoFormat** command on the **Format** menu in Design view.

On your own in Design view. You create a basic report and customize it in Design view to suit your requirements.

[How?](#)

1. In the [Database window](#), click **Reports**  under **Objects**.

2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click **Design View**.
4. Click the table or [query](#) that contains the data you want to base your report on. (If you want an [unbound report](#), don't select anything from this list.)



If you want to create a report that uses data from more than one table, base your report on a query.

5. Click **OK**.

Microsoft Access displays the report in [Design view](#).

Ways to customize a report

You can customize a report in the following ways:

Record source Change the tables and queries that a report is based on.

Sorting and grouping data You can sort data in ascending or descending order. You can also group records on one or more fields, and display subtotals and grand totals on a report.

Report window You can add or remove **Maximize** and **Minimize** buttons, change the title bar text, and other Report window elements.

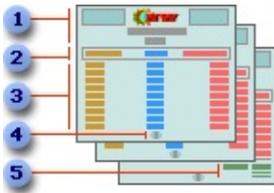
Sections You can add, remove, hide, or resize the header, footer, and details sections of a report. You can also set section properties to control the appearance and printing of a report.

Controls You can move, resize, or set the font properties of a control. You can also add controls to display calculated values, totals, current date and time, and other useful information on a report.

[Show All](#)

About sections of a report

The information in a report can be divided into sections. All reports have a [detail section](#), but a report can also include a [report header](#), a [page header](#), a [page footer](#), and a [report footer](#) section. Each section has a specific purpose and prints in a predictable order in the report.



- 1 Report header
- 2 Page header
- 3 Detail (data from tables)
- 4 Page footer
- 5 Report footer

In [Design view](#), sections are represented as bands, and each section that the report contains is represented once. In a printed report, some sections might be repeated many times. You determine where information appears in every section by placing [controls](#), such as [labels](#) and [text boxes](#).

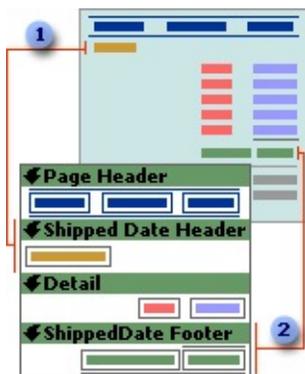


- 1 The report header appears once at the beginning of a report. You can use it for items such as a logo, report title, or print date. The report header is printed

before the page header on the first page of the report.

- 2 The page header appears at the top of every page in the report. You use it to display items such as column headings.
- 3 The detail section contains the main body of a report's data. This section is repeated for each record in the report's underlying record source.
- 4 The page footer appears at the bottom of every page in the report. You use it to display items such as page numbers.
- 5 The report footer appears once at the end of the report. You use it to display items such as report totals. The report footer is the last section in the report design but appears before the page footer on the last page of the printed report.

You can add a header and footer to each group in a report. In this report, orders shipped on the same date are grouped together. The header displays the value on which the records are grouped, and the footer displays a subtotal for the group.



- 1 A group header appears at the beginning of a new group of records. You use it to display information that applies to the group as a whole, such as a group name.
- 2 A group footer appears at the end of a group of records. You use it to display items such as group totals.

You can hide or resize a section, add a picture, or set the background color of a section. You can also set section properties to customize how the contents of a section print.

[Show All](#)

Add a page break to a report

 [Add a page break by using the page break control](#)

In reports, you use a page break control to mark where you want to start a new page within a section. For example, if you want a report's title page and introductory message printed on separate pages, place a page break in the [report header](#) after the controls that you want to appear on the title page and before the controls for the second page.

1. Open the report in [Design view](#).
2. Click the **Page Break** tool  in the [toolbox](#).
3. Click where you want to place the page break. Place the page break above or below a control to avoid splitting data in that control.

Microsoft Access marks the page break with a short dotted line on the left edge of the report.

Notes

- If you want each group or record in a report to start on a new page, set the **ForceNewPage** property of the [group header](#), [group footer](#), or [detail section](#).
- The Catalog report in the Northwind sample database has an example of using a page break in the report header. To view this report, open the Northwind database in your Microsoft Office folder's Samples folder, and then open the Catalog report in Design view.

 [Force a page break if a condition is met](#)

You can force a page break in a report if a condition is met by setting the **Visible** property of a page break control in a [macro](#) or an [event procedure](#).

1. Open the report in [Design view](#).
2. Click the **Page Break** tool  in the [toolbox](#).
3. Click where you want to place the page break.

4. Do one of the following:

[Create a macro](#)

1. Click **Properties**  on the toolbar, and specify a name for the page break control in the **Name** box.
2. In the [Database window](#), click **Macros**  under **Objects**.
3. Click the **New** button on the Database window toolbar.
4. In a blank [action row](#), click **SetValue** in the [action list](#).
5. Set the **Item** argument to the [identifier](#) for the **Visible** property of the page break control. For example, if the name of the control is CondPgBreak, set the **Item** argument to **[CondPgBreak].[Visible]**.
6. Set the **Expression** argument to **No**.
7. Click **Save**  to save the macro.
8. In report [Design view](#), set the **OnFormat** property of the report's [page header](#) section to the name of the macro.
9. This hides the page break control when the report starts formatting each page, so the page doesn't break.
10. Create a second macro that sets the **Visible** property of the page break control to **Yes** when a [conditional expression](#) is met. As you can see from the following example, to make a page break occur if the Counter control in the [detail section](#) is 10, set the detail section's **OnFormat** property to the name of the second macro.

When this condition is met...

Macro1 : Macro		
Condition	Action	Comm
▶ [Counter]=10	SetValue	

Action Arguments

Item	[CondPgBreak].[Visible]
Expression	Yes

... the macro sets the page break control's Visible property to Yes.

11. Click **Save** to save the macro.

12. Select the section where you placed the page break, and set the section's **OnFormat** property to the name of the second macro. In the preceding example, to make a page break occur if the Counter control in the detail section is 10, set the detail section's **OnFormat** property to the name of the second macro.
13. When the [condition](#) is met, the page breaks. After the page is broken, the macro attached to the page header hides the page break control until the condition is met again.

[Create an event procedure](#)

1. Double-click the section selector of the page header.
2. On the **Event** tab in the property sheet, click the **OnFormat** property.
3. Click **Build**  next to the property box to display the **Choose Builder** dialog box.
4. Double-click **Code Builder** to display the event procedure **Sub** and **End Sub** statements in the [report module](#).
5. In the event procedure, add an assignment statement that sets the **Visible** property of the page break control to **No**. For example, if the name of the control is CondPgBreak, add the following assignment statement:

```
Me![CondPgBreak].Visible = False
```

This hides the page break control when the report starts formatting each page, so the page doesn't break.

6. In the Format event procedure of the section where you placed the page break, add [Microsoft Visual Basic](#) code that sets the **Visible** property to **Yes** when a [condition](#) is met. For example, to make the [detail section](#) break when the value of the Counter control reaches 10, add the following code to the Detail_Format event procedure:

```
If Me![Counter] = 10 Then  
    Me![CondPgBreak].Visible = True  
End If
```

[Show All](#)

Reset page numbers for each group in a report

Before you do the following procedure, create a [macro](#) that uses the SetValue action to change the **Page** property to **1**.

1. Do one of the following:

If the page number is located in the [page header](#), use the following values for the SetValue [action arguments](#):

Item: **[Page]** and Expression: **0**

If the page number is located in the [page footer](#), use the following values for the SetValue action arguments:

Item: **[Page]** and Expression: **1**

2. Save and close the macro.
3. Open the report in [Design view](#).
4. Add a text box to show page numbers in the page header or footer.

[How?](#)

1. Open the form or report in [Design view](#).
2. On the **Insert** menu, click **Page Numbers**.
3. In the **Page Numbers** dialog box, select the format, position, and alignment for the page number. For alignment, the following options are available:

Left Page number shows at the left margin.

Center Page number is centered between the left and right margins.

Right Page number shows at the right margin.

Inside Odd page numbers print on the left, and even page numbers print on the right.

Outside Even page numbers print on the left and odd page numbers print on the right.

4. To show a number on the first page, select the **Show Number On First Page** check box.

Note Microsoft Access adds a [text box](#) to either the page header or the page footer based on your selection. The **ControlSource** property is set to an expression based on the format you chose. You can move, size, or position the text box, and set its properties to customize its appearance.

5. Do one of the following:

If the page number appears in the page header, set the **ForceNewPage** property of the first-level [group header](#) to **Before Section** and the **OnFormat** property of the first-level [group footer](#) to the name of the macro you created in step 1.

If the page number appears in the page footer, set the **OnFormat** property of the first-level group header to the name of the macro you created in step 1 and the **ForceNewPage** property of the first-level group footer to **After Section**.

Note The Employee Sales by Country report in the Northwind sample database uses an event procedure in the **OnFormat** property to reset the page number for each country/region. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Employee Sales by Country report in Design view.

[Show All](#)

About combining and manipulating text values

Combining text values

You can combine text values from multiple fields or [controls](#) by using the text concatenation operator (&) in an [expression](#). The concatenation operator concatenates two text strings and returns a single string.

For example, to return an address like:

507 20th Ave. E., Seattle WA 98122

type this expression:

```
=[Address]&" "&[City]&" "&[Region]&" "&[PostalCode]
```



[Combining text values when some fields might not contain data](#)

When you combine text values from several fields, some records might contain empty fields, resulting in output that might not be what you want. For example, the output might include extra spaces.

To compensate for records that don't have values in certain fields, you can use the **IIf** function. You use the **IIf** function when you want the value (or absence of a value) in one field to determine which one of two possible results you want returned.

For example, if you know that not every record will have a value in the Region field, use the following expression:

```
=IIf(IsNull([Region]),"", " "&[Region])
```

If the Region field is empty, Microsoft Access returns an empty string, as specified by the argument "". If the Region field contains a value, Microsoft Access returns a space and the region, as specified by the argument " " &

[Region].

Note The Invoice report in the Northwind sample database has an example of combining values from text fields by using the **IIf** function. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Invoice report in [Design view](#).

Manipulating text values

You can manipulate a text string by using a function, such as **Left** or **Trim**. For example, set the **ControlSource** property of a text box to the following expression to display the value of the Address field without any leading or trailing spaces.

```
=Trim([Address])
```

[Show All](#)

Count the number of records in each group or report

1. Open the report in [Design view](#).
2. Do one or both of the following:

[Count the number of records in a group](#)

1. Click the **Text Box** tool  on the [toolbox](#) and click the [detail section](#). The [control](#) will not be visible to the user, so its placement within the section is not critical.
2. Select the text box and click **Properties**  on the toolbar.
3. Set the following properties.

Property	Setting
Name	A text string, such as RecordCount
ControlSource	=1
RunningSum	Over Group
Visible	No

4. Add a text box to the [group footer](#).
5. Select the text box, and then click **Properties**  on the toolbar.
6. Set the **ControlSource** property to the name of the control in the detail section that's tracking the running total; for example, =[RecordCount]

[Count the number of records in a report](#)

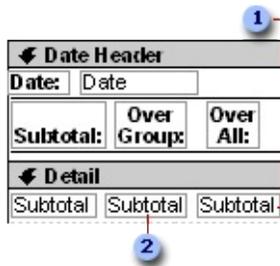
1. Click the **Text Box** tool  on the [toolbox](#) and click the [report header](#) or [footer](#) section where you want to place the text box.
2. Select the text box and click **Properties**  on the toolbar.
3. Set the **ControlSource** property of the text box to =**Count(*)**.

This [expression](#) uses the **Count** function to count all the records in the report even if some fields in some records are **Null**.

Example of a report that calculates running sums

You can calculate running sums that accumulate within each group and are reset each time a new group of records begins. You can also calculate running sums that accumulate throughout the report.

In the following illustration, for the two fields on the right that are bound to the Subtotal field, the **RunningSum** property is set to **Over Group** and **Over All**, respectively.



- 1 The **RunningSum** property of this text box is set to **Over All**.
- 2 The **RunningSum** property of this text box is set to **Over Group**.

In the following printed report, you can see how Microsoft Access calculates the running sums.

Date:			
Subtotal:	Over Group:	Over All:	
\$200	\$200	\$200	
\$350	\$550	\$550	
\$300	\$850	\$850	
Date:			
Subtotal:	Over Group:	Over All:	
\$250	\$250	\$1,100	
\$100	\$350	\$1,200	

- 1 The value in the Over Group column resets to zero each time a new group begins.
- 2 The value in the Over All column accumulates to the end of the report.

[Show All](#)

Calculate a running sum

1. Open the report in [Design view](#).
2. Click the **Text Box** tool  in the [toolbox](#).
3. Do one of the following:

To calculate a running sum that increases for each record, place the text box in the [detail section](#).

To calculate a running sum that increases for each group of records, place the text box in the [group header](#) or [footer](#).

4. Select the text box, and then click **Properties**  on the toolbar.
5. Set the **ControlSource** property to an expression that calculates the sum you want.
6. Set the **RunningSum** property according to the type of running total you want:

Over Group Resets to 0 (zero) at the beginning of each higher [group level](#).

Over All Accumulates until the end of the report.

Note When you set the **RunningSum** property to **Over All**, you can repeat the grand total in the [report footer](#). Create a text box in the report footer and set its **ControlSource** property to the name of the text box that calculates the running sum; for example, =[OrderAmount]

[Show All](#)

Number each detail record in a group or report

1. Open the report in [Design view](#).
2. Click the **Text Box** tool  on the [toolbox](#) and click the [detail section](#) where you want to add the text box for the number.
3. Select the text box and click **Properties**  on the toolbar.
4. Set the **ControlSource** property to =1.
5. Do one of the following:

To number each record and reset the counter to 1 for each group, set the **RunningSum** property to **Over Group**.

To number each record without resetting the counter for each group, set the **RunningSum** property to **Over All**.

Note The Sales by Year report in the Northwind sample database has an example of numbered detail records. To view this report, open the Northwind database in the Samples subfolder of your Office folder, and then open the Sales by Year report in Design view.

[Show All](#)

Combine text values

1. Open the form, report, or [data access page](#) in [Design view](#).
2. Do one of the following:

Click the **Text Box** tool  in the [toolbox](#).

On a data access page, you can use a bound span control instead of a text box.

3. Click the section where you want to place the control.
4. Select the control, and then click **Properties**  on the toolbar.
5. In the **ControlSource** property box, type an [expression](#) that concatenates the values from the appropriate fields.

 [Tip](#)

In a form or report in a [Microsoft Access database](#) (.mdb), you can also click the **Build** button  to use the [Expression Builder](#) to create the expression.

 [Example that generates a complete mailing address](#)

For example, to return an address like

507 20th Ave. E., Seattle WA 98122

type this expression:

=[Address]&" "&[City]&" "&[Region]&" "&[PostalCode]

Note The Employee Sales by Country and Customer Labels reports in the Northwind sample database have examples of combining text values. To view one of these reports, open the Northwind database in the Samples subfolder of the Office folder, and then open the desired report in Design view.

[Show All](#)

About creating and customizing mailing labels

Mailing labels are a special type of Microsoft Access [report](#). You bind the labels to a table or a [query](#), and Access generates a label for each record in the underlying [record source](#).

The Customer Labels report in the Northwind sample database shows three-across mailing labels. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Customer Labels report in [Design view](#).

Creating mailing labels

You can create labels either by using the Label Wizard or in Design view. The wizard asks you detailed questions about the record source, fields, layout, and format you want and creates the labels based on your answers. The wizard also customizes the page setup options for printing the labels.

Although you can manually create the labels in Design view, setting the page setup options could be a lengthy procedure. It might be faster to use the wizard to create the labels and customize the visual appearance of the labels in Design view.

Customizing mailing labels

You customize a label report the same way you customize any other report. A label report has a [detail section](#) that contains one or more [text boxes](#). You can add or remove text boxes, move or resize them, set their font properties, and set other text box properties to make the label look the way you want. You can also modify the page setup options if the labels don't print correctly.

[Show All](#)

Create mailing labels

Before creating labels that print on a dot-matrix or tractor-feed printer, set the default printer and paper size.

1. In the [Database window](#), click **Reports** .
2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click **Label Wizard**.
4. Click the table or [query](#) that contains the data for the labels, and then click **OK**.
5. Follow the instructions in the wizard.

If the resulting label report doesn't look the way you want, you can customize it in [Design view](#).

[Show All](#)

Customize labels created with the Label Wizard

If you want to make major changes to the labels, it's better to create a new label report with the Label Wizard.

If you want to make minor adjustments to the labels, do one or more of the following:

 [Learn about formatting a control](#)

In [Design view](#) of a form, report, or [data access page](#), you can format a control in the following ways:

Size. You can resize a control by using the sizing handles, or you can resize a control to the height or width of another control. When you are in Design view of a form or report, you can use the ruler to help you resize controls. The width of the ruler's unit of measure is based on the regional settings in Microsoft Windows Control Panel.

Color, transparency, and special effects. You can change the color of a control or make it transparent. You can also emphasize the appearance of a control by using special effects. For example, you can specify that a control is raised, sunken, or etched. On a form or report, you can also specify that a control is shadowed or chiseled.

Borders. You can set the color and width of borders of controls. Also, you can set the line style of borders. For example, you can specify that a border consists of dashes or dots.

Appearance of text. You can specify the font and font size for text in a control. Also, you can make the text in a control bold, italic, or underlined.

Position. You can move a control or align it relative to another control. You can increase or decrease the space between controls, or you can specify that controls are evenly spaced.

About changing the properties of several controls at once

You can change property settings for a group of controls of the same type or for a group of controls of different types. If you select controls of different types, Microsoft Access displays only the properties that are shared by the group in the property sheet. If all the controls that you've selected share the same property setting, that setting appears in the property sheet; otherwise, that property box is blank. If you change the property setting, the change is applied to all selected controls.

Notes

- To select several controls at once in a data access page, you must have Microsoft Internet Explorer 5.5 or later installed on your computer.
- In a form or report, the **Name** property never appears when more than one control is selected because control names must be unique.

[Add a text box](#)

1. Open a report in [Design view](#).
2. Do one of the following:

[Create a bound text box](#)

Before creating a [bound](#) control, make sure that your report is bound to a [record source](#).

1. If the [field list](#) isn't visible, click **Field List**  on the [toolbar](#).
2. Select one or more fields in the field list on a report.
3. Drag the field or fields from the field list and position them on the report.

Access places one text box on the report for each field that you select

in the field list. Each text box is bound to one field in the underlying data source. Each text box also has an attached label by default.

Note Microsoft Access won't create the control if you try to add it by clicking the form, report, or data access page instead of dragging the selected field from the field list.

[Create an unbound text box](#)

1. Open a report in [Design view](#).
2. Click the **Text Box** tool  in the [toolbox](#).
3. Click anywhere on the report to create a default-sized text box, or drag to create a text box that's the size you want
3. Size the text box so that it's the appropriate size for the data you want to show.
4. Change the label text if necessary.
5. Switch to [Print Preview](#) to test the control.

[Move a control](#)

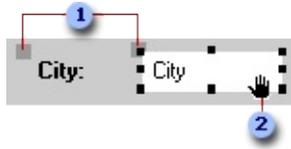
1. Open a form, report, or [data access page](#) in [Design view](#).
2. Do one of the following:

[Move a control and its label together](#)

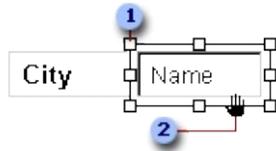
1. Click the [control](#) or its [label](#).

When you click either part of a [compound control](#) in forms and reports, Microsoft Access displays the move handles for both controls, as well as the sizing handles for the control that you clicked. On data access pages, when the control is selected, the label is not automatically selected. If you move the control, the label goes with it. But if you move the label, the control does not move.

2. Move the pointer over the border (not the move handle) of either the control or its label. After the pointer changes to an open hand, you can drag the control or controls to a new position.



- 1 Move handles of a control in a form or report.
- 2 When the pointer changes to an open hand, you can drag the control.



- 1 Move handle of a control in a data access page
- 2 When the pointer changes to an open hand, you can drag the control.

[Move a control and its label separately](#)

In a data access page, you can't move a control separately from its label, but you can select the label and move it separately from the control.

1. Click the [control](#) or its [label](#).
2. Do one of the following:
 - In a form or report, place the pointer over the move handle in the upper-left corner of the control or label, and the pointer changes to an upward-pointing hand.



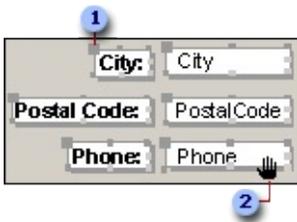
- 1 When the pointer changes to an upward-pointing hand, you can drag the control.
 - In a data access page, place the pointer on any edge of the label, and the pointer changes to an open hand.
3. Drag the control or label to its new position.

Notes

- The parts of a [compound control](#) remain related, even if you move the individual parts separately.
- To move an attached label to another [section](#) without moving the control, you must use the **Cut** and **Paste** commands. In a form or report, if you move a label to another section, the label will no longer be related to the control.

[Move two or more controls at the same time](#)

1. Select the controls by holding down the SHIFT key as you click each control.
2. Pass the pointer over the border (not the move handle) of any selected control until it turns into an open hand, and then drag the controls to a new position.



- 1 Move handle
- 2 When the pointer changes to an open hand, you can drag the controls.

[Move a control in front of or behind other controls](#)

- Click the [control](#) that you want to move, and then on the **Format** menu, click **Send to Back** or **Bring to Front**.

Note A list box or [subform](#) is always in front of other controls; you can't format a list box or subform so that it is behind another control.

[Tips](#)

- When you move a control in a form or report, you can maintain its horizontal or vertical alignment with other controls by holding down the SHIFT key while selecting and moving the control. You can move the control horizontally or vertically only, depending on which direction you

move it first.

- If you need to make small adjustments to the placement of a control, you might find it easier to press the appropriate arrow keys. If **Snap to Grid** is on when you use the arrow keys to move a control, Access aligns the control to the grid.
- If you want to temporarily override the current **Snap to Grid** setting, hold down the CTRL key while you're placing, moving, or sizing a control. For example, if **Snap to Grid** is turned on, you can hold down the CTRL key and move a control anywhere on the form, report or data access page without aligning it to the grid. Releasing the CTRL key restores the current **Snap to Grid** setting.

[Resize a control](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#) or controls that you want to resize. To select more than one control, hold down the SHIFT key and click each control, or drag a rectangle over the controls.
3. Drag the sizing handles until the control is the size you want.



- 1 Move handle
- 2 Sizing handles

If you select more than one control, all the controls change size as you drag the sizing handle of one control.

[Show All](#)

Create a form to enter report criteria

1. Create an [unbound form](#) that prompts for report [criteria](#).

[How?](#)

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Form** dialog box, click **Design View**, and click **OK**.
4. In [Design view](#), set the following form properties.

Property	Setting
----------	---------

Caption	Name you want to appear in the title bar of the form
----------------	--

DefaultView	Single Form
--------------------	--------------------

AllowFormView	Yes
----------------------	------------

AllowDatasheetView	No
---------------------------	-----------

AllowPivotTableView	No
----------------------------	-----------

AllowPivotChartView	No
----------------------------	-----------

ScrollBars	Neither
-------------------	----------------

RecordSelectors	No
------------------------	-----------

NavigationButtons	No
--------------------------	-----------

BorderStyle	Dialog
--------------------	---------------

5. Click the **Text Box** tool  on the [toolbox](#) to add a text box to the form for each criteria you want to enter.
6. Set the properties for the text boxes as follows.

Property	Setting
----------	---------

Name	Name that describes the type of criteria; for example, BeginningDate .
-------------	---

Format	Format that reflects the data type of the criteria. For example, for a date criteria, select a format such as Medium Date .
---------------	--

7. Save the form and give it a name, such as Sales Dialog.

You'll add OK and Cancel command buttons to the form after you create macros for them.

2. In the [Database window](#), click **Macros** , and then click the **New** button on the Database window toolbar.
3. Create a macro that opens the Sales Dialog form.

 [How?](#)

1. Begin by clicking **Macro Names**  to display the **Macro Name** column. Type a macro name, such as **Open Dialog**, in the **Macro Name** column, and then click the **OpenForm** [action](#). Then set the [action arguments](#) as follows.

Argument	Setting
Form Name	Sales Dialog
View	Form
Data Mode	Edit
Window Mode	Dialog

2. Add a second action, CancelEvent, that cancels previewing or printing the report when the Cancel button on the form is clicked.
3. If the **Condition** column is not displayed, click **Conditions**  .
4. Type the following [expression](#) in the **Condition** column:

Not IsLoaded("Sales Dialog")

Note IsLoaded is a function defined in the Utility Functions module in the Northwind sample database. It's used to check whether a form is open in [Form view](#) or [Datasheet view](#). You must define the IsLoaded function in your database before you can use it. (You can copy and paste this function from Northwind into a utility module in your database.)

4. Create a macro that closes the form.

 [How?](#)

Give the macro a name, such as Close Dialog. Click the Close [action](#). Then set its [action arguments](#) as follows:

Argument	Setting
Object Type	Form
Object Name	Sales Dialog
Save	No

5. Create a macro for the OK button.



This macro hides the form. Give the macro a name, such as OK, and click the SetValue [action](#). Then set its [action arguments](#) as follows:

Argument	Setting
Item	[Visible]
Expression	No

6. Create a macro for the Cancel button.



This macro closes the form. Give the macro a name, such as Cancel, and click the Close [action](#). Then set its [action arguments](#) as follows:

Argument	Setting
Object Type	Form
Object Name	Sales Dialog
Save	No

7. Save and close the macro group. Give the macro group a name— for example, the same name that you gave the unbound form.
8. Add OK and Cancel command buttons to the form.



1. Reopen the Sales Dialog form in [Design view](#).

2. Make sure **Control Wizards**  in the [toolbox](#) isn't selected and create an OK command button.
3. Set its properties as follows.

Property	Setting
----------	---------

Name	OK
-------------	-----------

Caption	OK
----------------	-----------

Default	Yes
----------------	------------

OnClick	Name of the macro; for example, Sales Dialog.OK
----------------	---

4. Create a Cancel command button, and set its properties as follows.

Property	Setting
----------	---------

Name	Cancel
-------------	---------------

Caption	Cancel
----------------	---------------

OnClick	Name of the macro; for example, Sales Dialog.Cancel
----------------	---

5. Save and close the form.
9. Enter the criteria in the underlying query or stored procedure for the report.

[How?](#)

1. Open the underlying query or stored procedure for the report in [Design view](#).
2. Enter the criteria for the data. In the [expression](#), use the **Forms** object, the name of the form, and the names of the [controls](#) in the criteria.
 - For example, in a [Microsoft Access database](#) (.mdb), for a form called Sales Dialog, you would use the following expression to refer to controls named Beginning Date and Ending Date in the query:

Between [Forms]![Sales Dialog]![Beginning Date] And [Forms]![Sales Dialog]![Ending Date]

- In a [Microsoft Access project](#) (.adp), you must first explicitly name the parameters in the stored procedure; for example:

@Beginning_Date datetime, @Ending_Date datetime

and then use those parameters in the WHERE clause; for example:

WHERE Sales.ShippedDate Between @Beginning_Date And @Ending_Date

In an Access project, you set the reference to the controls on the form in the **InputParameters** property in the report, as shown in the next procedure.

10. In a [Microsoft Access project](#), set the **InputParameters** property in the main report.

[How?](#)

1. Open the report in [Design view](#).
2. Set the report's **InputParameters** property to a string that specifies the parameters that are passed to the stored procedure that the report is bound to. As in the following example, the string must be an [expression](#) that includes the parameters you specified in the stored procedure and the reference to the controls on the dialog box:

```
@Beginning_date datetime = [Forms]![Sales Dialog]![Beginning Date], @Ending_date datetime = [Forms]![Sales Dialog]![Ending Date]
```

11. Attach the macros to the main report.

[How?](#)

1. Open the report in [Design view](#).
2. Set the following report properties.

Property	Setting
OnOpen	Name of the macro that opens the Sales Dialog form; for example, Sales Dialog.Open Dialog
OnClose	Name of the macro that closes the form; for example, Sales Dialog.Close Dialog

[Show All](#)

Display report criteria in a report

1. Do one of the following:
 - In a [Microsoft Access database \(.mdb\)](#), create the criteria parameters in the report's underlying query.
 - In a [Microsoft Access project \(.adp\)](#), create the parameters in the report's underlying [stored procedure](#) or user-defined function.
2. Open the report in [Design view](#).
3. For each parameter, add a text box by clicking the **Text Box** tool  in the [toolbox](#), and then clicking where you want to place it.
4. Click **Properties**  on the toolbar to display the [property sheet](#) for the text box.
5. Do one of the following:

In an Access database, set the **ControlSource** property for the text box to the parameters you specified in the **Criteria** cell in the query.

 [Example of setting the **ControlSource** property to one or more parameters in an Access database](#)

If the parameters in the query are:

Between [Type a beginning date] And [Type an ending date]

and you want to display them in two text boxes, set the **ControlSource** property of one text box to:

[Type a beginning date]

and the **ControlSource** property of the other text box to:

[Type an ending date]

If you want to display both parameters in one text box, you can use an [expression](#) in the **ControlSource** property box; for example:

= "Between " & [Type a beginning date] & " and " & [Type an ending date]

In an Access project, set the report's **InputParameters** property to a string that specifies the parameters that are passed to the stored procedure that the report is bound to. Then set the **ControlSource** property of the text box to a string that specifies the parameters in the stored procedure, any text you want to include, and the prompt text enclosed in square brackets.

 [Example of setting the report and text box properties in an Access project](#)

If the parameters in the query are:

Between [Type a beginning date] And [Type an ending date]

The string you assign to the **InputParameters** property must be an [expression](#) that includes the parameters you specified in the stored procedure or user-defined function and the text you want to appear in the prompt; enclose the prompt text in square brackets ([]):

@Beginning_date datetime = [Beginning Date], @Ending_date datetime = [Ending Date]

Then set the **ControlSource** property of the text box as shown in the following example:

= "Sales from " & [Beginning Date] & " to " & [Ending Date]

6. To see the criteria, print or preview the report. Microsoft Access displays the parameter values exactly as you typed them in the prompt.

Note The Employee Sales by Country report in the Northwind sample database prints the criteria in the [report header](#). To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Alphabetical List of Products report in Design view.

[Show All](#)

Hide duplicate data in a report

[Hide duplicate data on a report with only sorted records](#)

1. Open a sorted report that might contain duplicate values in [Design view](#).
2. Select the text box that might contain duplicate data and click **Properties**  on the toolbar.
3. Set the **HideDuplicat**es property to **Yes**.

In a text box for which **HideDuplicat**es is set to **Yes**, Microsoft Access prints a value in the first record in which that value occurs and, if records with the duplicate value continue onto another page, at the top of a new page.

[Hide duplicate data on a report with grouped records](#)

1. Open a grouped report that might contain duplicate values in [Design view](#).
2. Move the text box for the field you're grouping on from the [group header](#) to the [detail section](#).
3. Select the text box and click **Properties**  on the toolbar.
4. Set the **HideDuplicat**es property to **Yes**.
5. Remove lines and all other [controls](#) from the group header.
6. Double-click the [section selector](#) of the group header to display its [property sheet](#).
7. Set the **Height** property of the group header to **0**.

The group name will appear only at the beginning of a new group or at the top of a new page if the group continues for more than one page.

[Show All](#)

Examples of reports that start each group in a new row or column

In multiple-column reports, you can emphasize group headings by placing them either down the left side of the page or across the top of the page.

 [Example of a report that starts each group in a new row](#)



- 1 In this report, each [group header](#) prints in a new row in the left column ...
- 2 ... and the detail records print in the next row of the left column, in columns across and then down the page.

The text box for CategoryID is placed in the group header and the text box for ProductName is placed in the [detail section](#). The **NewRowOrCol** property of the CategoryID header is set to **Before & After**. The properties on the **Columns** tab in the **Page Setup** dialog box are set as follows:

Group	Property	Setting
Grid Settings	Number Of Columns	3
Column Size	Width	The width of one column in the detail section; for example, 2"
	Height	The height of one row; for example, 0.25"
Column Layout	Across, Then Down	Selected

 [Example of a report that starts each group in a new column](#)

The diagram shows a report layout with three columns. The first row contains three group headers: 'Category1', 'Category2', and 'Category3'. A bracket labeled '1' spans these three headers. Below the headers are detail records. A bracket labeled '2' spans the first three records: 'Chai', 'Côte de Blaye', and 'Ipoh Coffee'. The second record, 'Côte de Blaye', is split across two columns. The third record, 'Ipoh Coffee', is also split across two columns. The fourth record, 'Lakkalikööri', is split across two columns. The fifth record, 'Sirop d'érable', is split across two columns. The sixth record, 'Vegie-spread', is split across two columns.

Category1	Category2	Category3
Chai	Anis Syrup	Chocolate
Côte de Blaye	Genen Shouyu	Maxilaku
Ipoh Coffee	Gula Malacca	Pavalova
Lakkalikööri	Sirop d'érable	
	Vegie-spread	

- 1 In this report, each [group header](#) prints at the top of a column ...
- 2 ... and the detail records print down, and then across the page starting in the next row under the group header.

The design of this report is similar to the one in the previous example, but the **NewRowOrCol** property of the CategoryID header is set to **Before Section**, and **Column Layout** is set to **Down, Then Across** in the **Page Setup** dialog box.

[Show All](#)

About subreports

A subreport is a report that's inserted in another report. When you combine reports, one of them must serve as the main report. A main report is either bound or unbound; that is, it is or isn't based on a table, [query](#), or [SQL statement](#).

[Example of an unbound main report with two unrelated subreports](#)

An unbound main report can serve as a container for unrelated subreports that you want to combine.



Employee Sales:	Employee Name:	Employee Sales:
	██████████	\$93,932
	██████████	\$88,123
	██████████	\$123,123

Category Sales:	Category Name:	Category Sales:
	██████████	\$127,183
	██████████	\$56,462
	██████████	\$88,500

- 1 The unbound main report contains two subreports.
- 2 One subreport summarizes sales by employee.
- 3 The other subreport summarizes sales by category.

[Example of a main report and subreport that are bound to the same record source](#)

You can use the main report to show detail records, such as every sale in a year, and then use a subreport to show summary information, such as the total sales for each quarter.

Quarter:	Orders:	Sales:
1	31	\$142,758
2	30	\$140,645
3	108	\$147,028
4	108	\$176,567
Totals:	397	\$606,998

Shipped Date:	Order ID:	Sales:
03-Jan	10396	\$1,903
05-Jan	10399	\$1,765
	10404	\$1,531

- 1 The subreport summarizes the year's sales by quarter.
- 2 The main report lists the day-to-day sales.

 [Example of a main report and a subreport that are bound to related record sources](#)

A main report can contain data common to two or more parallel subreports. In this case, the subreport contains data that is related to the data in the main form.

Food Fairs	
n/n/nn	
Gourmet Expo	
Representatives:	
[Redacted]	
[Redacted]	
International Food Fair	
Representatives:	
[Redacted]	
[Redacted]	

- 1 The main report lists the name and city of each fair.
- 2 The subreport lists the representatives who will attend each fair.

Subforms in a report

A main report can include [subforms](#) as well as subreports, and it can include as many subforms and subreports as you want. Additionally, a main report can contain up to two levels of subforms and subreports. For example, a report can contain a subreport, and that subreport can contain a subform or a subreport. The following table shows the possible combinations of subforms and subreports in a main report.

Level 1	Level 2
Subreport 1	Subreport 2
Subreport 1	Subform 1
Subform 1	Subform 2

Synchronizing a report and subreport

When you insert a subreport that contains information related to data in the main report, the [subreport control](#) must be linked to the main report. The link ensures that the records printed in the subreport correspond correctly to the records printed in the main report.

When you create a subreport by using a wizard or by dragging a report or datasheet from the [Database window](#) to another report, Microsoft Access automatically synchronizes the main report and subreport if:

- You have defined [relationships](#) for the tables you select, or you have defined relationships for the tables underlying the queries you select.
- The main report is based on a table with a [primary key](#) and the subreport is based on a table that contains a field with the same name as that primary key and with the same or a compatible [data type](#). For example, if the primary key of the table underlying the main form is an AutoNumber field and its **FieldSize** property is set to **Long Integer**, the corresponding field in the table underlying the subform should be a Number field with its **FieldSize** property set to **Long Integer**. If you select a query or queries, the underlying tables in the query or queries must meet the same conditions.

Note The Sales by Category and Sales by Year reports in the Northwind sample database use subreports. To view these reports, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the report in [Design view](#).

[Show All](#)

Create a subreport

If the subreport will be linked to the main report, make sure the underlying [record sources](#) are [related](#) before using this procedure.

[Create a subreport in an existing report](#)

1. Open the report you want to use as the main report in [Design view](#).
2. Make sure the **Control Wizards** tool  in the [toolbox](#) is selected.
3. Click the **Subform/Subreport** tool  in the toolbox.
4. On the report, click where you want to place the subreport.
5. Follow the directions in the wizard dialog boxes.

When you click the **Finish** button, Microsoft Access adds a [subreport control](#) to your report. It also creates a separate report that is displayed as the subreport.

[Add an existing report to another report as a subreport](#)

1. Open the report you want to use as the main report in [Design view](#).
2. Press F11 to switch to the [Database window](#).
3. Drag a report or datasheet from the Database window to the section you want the subreport to appear in on the main report.

Microsoft Access adds a [subreport control](#) to the report.

[Show All](#)

Work with subreports

[Learn about subreports](#)

A subreport is a report that's inserted in another report. When you combine reports, one of them must serve as the main report. A main report is either bound or unbound; that is, it is or isn't based on a table, [query](#), or [SQL statement](#).

[Example of an unbound main report with two unrelated subreports](#)

An unbound main report can serve as a container for unrelated subreports that you want to combine.

Employee Sales:	Employee Name:	Employee Sales:
		\$93,932
		\$88,123
		\$123,123

Category Sales:	Category Name:	Category Sales:
		\$127,183
		\$56,462
		\$88,500

- 1 The unbound main report contains two subreports.
- 2 One subreport summarizes sales by employee.
- 3 The other subreport summarizes sales by category.

[Example of a main report and subreport that are bound to the same record source](#)

You can use the main report to show detail records, such as every sale in a year, and then use a subreport to show summary information, such as the total sales for each quarter.

Quarter:	Orders:	Sales:
1	91	\$142,758
2	90	\$140,645
3	108	\$147,028
4	108	\$176,567
Totals:	397	\$606,998

Shipped Date:	Order ID:	Sales:
03-Jan	10396	\$1,903
05-Jan	10399	\$1,765
	10404	\$1,591

- 1 The subreport summarizes the year's sales by quarter.
- 2 The main report lists the day-to-day sales.

 [Example of a main report and a subreport that are bound to related record sources](#)

A main report can contain data common to two or more parallel subreports. In this case, the subreport contains data that is related to the data in the main form.

Food Fairs	
n/n/nn	
Gourmet Expo	
Representatives:	
[Redacted]	
[Redacted]	
International Food Fair	
Representatives:	
[Redacted]	
[Redacted]	

- 1 The main report lists the name and city of each fair.
- 2 The subreport lists the representatives who will attend each fair.

Subforms in a report

A main report can include [subforms](#) as well as subreports, and it can include as many subforms and subreports as you want. Additionally, a main report can contain up to two levels of subforms and subreports. For example, a report can contain a subreport, and that subreport can contain a subform or a subreport. The following table shows the possible combinations of subforms and subreports in a main report.

Level 1	Level 2
Subreport 1	Subreport 2
Subreport 1	Subform 1
Subform 1	Subform 2

Synchronizing a report and subreport

When you insert a subreport that contains information related to data in the main report, the [subreport control](#) must be linked to the main report. The link ensures that the records printed in the subreport correspond correctly to the records printed in the main report.

When you create a subreport by using a wizard or by dragging a report or datasheet from the [Database window](#) to another report, Microsoft Access automatically synchronizes the main report and subreport if:

- You have defined [relationships](#) for the tables you select, or you have defined relationships for the tables underlying the queries you select.
- The main report is based on a table with a [primary key](#) and the subreport is based on a table that contains a field with the same name as that primary key and with the same or a compatible [data type](#). For example, if the primary key of the table underlying the main form is an AutoNumber field and its **FieldSize** property is set to **Long Integer**, the corresponding field in the table underlying the subform should be a Number field with its **FieldSize** property set to **Long Integer**. If you select a query or queries, the underlying tables in the query or queries must meet the same conditions.

Note The Sales by Category and Sales by Year reports in the Northwind sample database use subreports. To view these reports, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the reports in Design view.



[Move a subreport](#)

1. Click the subreport control.
2. Place the pointer over the move handle in the upper-left corner of the control, and the pointer changes to an upward-pointing hand.
3. Drag the control or label to its new position.



[Resize a subreport](#)

1. Click the subreport control.

2. Drag the sizing handles until the control is the size you want.

[Show All](#)

Create a crosstab report

1. Create a crosstab query.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Crosstab Query Wizard**, and then click **OK**.
3. Follow the directions in the wizard dialog boxes. In the last dialog box, you can choose to run the query or see the query's structure in [Design view](#).

Note If the resulting query isn't exactly what you want, you can rerun the wizard or change the query in Design view.

2. Open the query in [Design view](#).
3. Display the query's [property sheet](#) by double-clicking anywhere outside the [design grid](#) and outside the field lists.
4. In the **ColumnHeadings** property box, type the column headings you want to appear in the report, separating each one with a semicolon. If a column heading contains spaces, enclose it in double (") quotation marks.

For instance, in a query showing sales by quarter, you would type:

"Qtr 1";"Qtr 2";"Qtr 3";"Qtr 4"

5. Create a blank report that is bound to the crosstab query.

[How?](#)

1. In the [Database window](#), click **Reports**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Report** dialog box, click **Design View**.
4. Select the crosstab query, and then click **OK**.

6. Add the [controls](#) for the fields you want to appear in the report.
7. Place labels for the row headings and column headings in the [page header](#).
8. Place text boxes for the row headings and column values in the [detail section](#).

[Show All](#)

Customize a multiple-column report

Do one or more of the following:

[Start each group in a new row or column](#)

1. Open a multiple-column report that has at least one [group level](#) in [Design view](#).
2. Click **Page Setup** on the **File** menu.
3. Click the **Columns** tab. Under **Grid Settings**, make sure **Number of Columns** is set to a value greater than 1.
4. Under **Column Layout**, do one of the following:

To start each group in a new column, click **Down, Then Across**.

To start each group in a new row, click **Across, Then Down**.

5. Click **OK**.
6. Double-click the [section selector](#) for the [group header](#) or [footer](#) to display its [property sheet](#).
7. Set the **NewRowOrCol** property to one of the following settings:

None (Default) The row or column breaks are determined by the setting in the **Page Setup** dialog box and by the available space on the page.

Before Section Microsoft Access starts printing the current section (the section for which you're setting the property, such as the group header) in a new row or column. It starts printing the next section, for example, the [detail section](#), in the same row or column.

After Section Microsoft Access starts printing the current section, such as the group header, in the current row or column. It starts printing the next section, such as the detail section, in the next row or column.

Before & After Microsoft Access starts printing the current section, such as the group header, in a new row or column. It starts printing the next section, such as the detail section, in the next row or column.

Note The Products by Category report in the Northwind sample database starts each group in a new column. To view this report, open the Northwind database in the Samples subfolder of your Office folder, and then open the Products by Category report in Design view.

[Keep a group together in a column](#)

1. Open a multiple-column report that has at least one [group level](#) in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And Grouping** box.
3. Click the field or [expression](#) for the group level that you want to keep together.
4. In the group properties section of the **Sorting And Grouping** box, set the **KeepTogether** property to one of the following settings:

Whole Group. The [group header](#), all detail records, and the [group footer](#) are printed in the same column.

With First Detail. The group header is printed in a column only if it can also print the first detail record.

5. Double-click the report selector.
6. Set the **GrpKeepTogether** property to **Per Column/Row**.

[Repeat a group header in another column](#)

1. Open the report in [Design view](#).
2. Double-click the [section selector](#) of the [group header](#) to display the property sheet.
3. Set the **RepeatSection** property to **Yes**.

[Keep a section together in a column](#)

1. Open the report in [Design view](#).
2. Double-click the [section selector](#) to open the [property sheet](#).
3. Set the **KeepTogether** property to **Yes**.

[Show All](#)

Create a multiple-column report

1. Create a report and open it in Design view.
2. On the **File** menu, click **Page Setup**.
3. In the **Page Setup** dialog box, click the **Columns** tab.
4. Under **Grid Settings**, in the **Number Of Columns** box, type the number of columns you want on each page.
5. In the **Row Spacing** box, type the amount of vertical space you want between each record in the detail section.

Note If you left space between the last [control](#) in the detail section and the bottom edge of the detail section, you can leave **Row Spacing** set to **0**.

6. In the **Column Spacing** box, type the amount of space you want between the columns.
7. Under **Column Size**, type the width you want for a column in the **Width** box. You can set the height of the detail section by typing a number in the **Height** box or by adjusting the height of the section in [Design view](#).
8. Under **Column Layout**, click **Down, Then Across** or **Across, Then Down**.
9. Click the **Page** tab.
10. Under **Orientation**, click **Portrait** or **Landscape**.

Notes

- When a multiple-column report is printed, the [report header](#) and [footer](#) and [page header](#) and [footer](#) span the full width of the report, but the [group header](#) and [footer](#) and [detail section](#) of a multiple-column report span the width of one column.
- The Products by Category report in the Northwind sample database displays data in multiple columns. To view this report, open the Northwind database in the Samples subfolder of your Office folder, and then open the Products by Category report in Design view.

[Show All](#)

About report snapshots and Snapshot Viewer

A report snapshot is a file (.snp extension) that contains a high-fidelity copy of each page of a Microsoft Access [report](#) and that preserves the two-dimensional layout, graphics, and other embedded objects of the report.

Snapshot Viewer is a program that you can use to view, print, and mail a snapshot, such as a report snapshot. Snapshot Viewer version 11.0 consists of a stand-alone executable program, a [Snapshot Viewer control](#) (snapview.ocx), a Help file, and other related files. By default, Snapshot Viewer is automatically installed by Access the first time you create a report snapshot. You can also install Snapshot Viewer from the Setup program, or from a software download page located on the Microsoft Office Developer's Web site.

You can use the Snapshot Viewer control to view a snapshot from Microsoft Internet Explorer version 3.0 or later or from any application that supports [ActiveX controls](#), such as Access or [Microsoft Visual Basic for Applications](#).

You use Access to create a report snapshot. However, you can view, print, store, publish, distribute, and archive a report snapshot without having an Access license by using a combination of Snapshot Viewer and other programs such as Microsoft Windows Explorer, an e-mail program, or a Web browser, such as Microsoft Internet Explorer.



[Benefits of report snapshots and Snapshot Viewer](#)

If you print and distribute reports to users inside or outside your organization, using report snapshots can save considerable time and money. Instead of photocopying and mailing printed Access reports, you can electronically distribute and publish them using either e-mail or a Web browser. Your users can easily and quickly view the reports online and print only the report pages they need. This is especially useful when your reports contain color and images, such as charts and pictures.

[Using report snapshots on the World Wide Web](#)

There are two ways you can publish a report snapshot to the [World Wide Web](#):

- Create a link to the report snapshot by using an [HTML](#) anchor tag on a Web page

If you are using Internet Explorer version 3.0 or later as your Web browser, when you click the link, Internet Explorer does an in-place activation of the report snapshot and Snapshot Viewer in the Internet Explorer window. Snapshot Viewer must be installed on the same computer as the browser. Therefore, it's a good idea to provide an additional link on the Web page to the software download page for Snapshot Viewer. Users who do not have Snapshot Viewer installed on their computers can then download and install Snapshot Viewer to view the report.

- Embed the Snapshot Viewer control on the Web page by using the HTML Object tag

You can embed the Snapshot Viewer control by using the Microsoft ActiveX Control Pad application, which adds an HTML Object tag, and set the control parameters by using the Control Pad Object Editor.

You can use this method only with Internet Explorer 3.0 or later. In this case, if Snapshot Viewer is not installed on the same computer as the browser, Internet Explorer automatically downloads the Snapshot Viewer ActiveX control (but not the entire Snapshot Viewer program).

[Sending report snapshots in e-mail](#)

You can send a report snapshot in e-mail by using Snapshot Viewer or Access. You can also embed a report snapshot as an icon or as a file in an e-mail message by using an e-mail program that supports ActiveX controls, such as Microsoft Outlook or Microsoft Exchange. When you send report snapshots in e-mail, it's a good idea to include a link to the software download page for Snapshot Viewer.

[Automatically creating and distributing report snapshots](#)

A typical business application creates and distributes reports not only on request,

but also on a schedule according to regular business cycles, such as weekly stock quotes, monthly sales reports, or quarterly financial statements.

From Access, you can programmatically create and distribute report snapshots in several ways. For example:

- Create a [macro](#) that uses the OutputTo action to export one or more reports in snapshot format to a Web server that you use to publish your reports.
- Create a macro that uses the SendObject action to mail a report to a mail recipient or distribution list.
- Create a Visual Basic for Applications program by using Snapshot Viewer methods, events, and properties.

[Show All](#)

Create a report snapshot

1. In the [Database window](#), click the name of the report you want to create a report snapshot from.
2. On the **File** menu, click **Export**.
3. In the **Save As Type** box, click **Snapshot Format (*.snp)**.
4. Click the arrow to the right of the **Save In** box, and select the drive or folder to export to.
5. In the **File Name** box, enter the file name.
6. Click **Save**.

If the report contains a [parameter query](#), Microsoft Access first prompts you for the parameter values and then outputs the results.

By default, the [report snapshot](#) is displayed in [Snapshot Viewer](#). If this is the first time you are creating a report snapshot and you chose not to install Snapshot Viewer during Setup, Access automatically installs Snapshot Viewer for you.

[Show All](#)

Open a report snapshot

1. Start Windows Explorer.
2. Locate the folder that contains the [report snapshot](#).
3. Double-click the report snapshot icon.

[Show All](#)

Publish a report snapshot to the Web

1. Create a [macro](#).

[How?](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
 2. In the **Action** column, click the arrow to display the [action list](#).
 3. Click the action you want to use.
 4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
 5. Type a comment for the action. Comments are optional.
 4. To add more actions to the macro, move to another [action row](#) and repeat step 3. Microsoft Access carries out the actions in the order you list them.
2. Add the **OutputTo** action to the macro. Make sure you output directly to the folder on the Web server. For example:

Argument	Value
Object Type	Report
Object Name	Catalog
Output Format	Snapshot Format
Output File	\\Website\Inetpub\wwwroot\Webreports\Catalog.snp

3. Run the macro each time you want to export the [report snapshot](#).

How?

To run a macro directly, do one of the following:

- To run a macro from the [Macro window](#), click **Run**  on the [toolbar](#).
 - To run a macro from the [Database window](#), click **Macros**  and then double-click a macro name.
 - On the **Tools** menu, point to **Macro**, click **Run Macro**, and then select the macro from the **Macro Name** list. Then click a macro in the **Macro Name** box.
 - Run a macro from a [Microsoft Visual Basic](#) procedure by using the **RunMacro** method of the **DoCmd** object.
4. On the home page of the Web site, do one of the following:

Create a link to the report snapshot using an [HTML](#) anchor tag. For example:

```
<A HREF="http://Website/WebReports/catalog.snp">Catalog report</
```

Embed the [Snapshot Viewer control](#) in an HTML Object tag by using the Microsoft ActiveX Control Pad application, and set the control parameters using the Control Pad Object Editor (you can only use this method with Microsoft Internet Explorer version 3.0 or later). For example:

```
<OBJECT ID="SnapshotViewer" WIDTH=640 HEIGHT=480
  CLASSID="CLSID:F0E42D60-368C-11D0-AD81-00A0C90DC8D9">
  <PARAM NAME="_ExtentX" VALUE="16722">
  <PARAM NAME="_ExtentY" VALUE="11774">
  <PARAM NAME="_Version" VALUE="65536">
  <PARAM NAME="SnapshotPath"
  VALUE="http://Website/WebReports/catalog.snp">
  <PARAM NAME="Zoom" VALUE="0">
  <PARAM NAME="AllowContextMenu" VALUE="-1">
  <PARAM NAME="ShowNavigationButtons" VALUE="-1">
</OBJECT>
```

When you click the link or the Snapshot Viewer control, Internet Explorer displays the report snapshot.

Note If the report is based on a [parameter query](#), modify the macro to run it in a batch file. You can use the SendKeys action to pass the parameter information, or you can modify the report by removing the parameter query and using specific criteria.

Security Avoid using the **SendKeys** statement or an AutoKeys macro with sensitive or confidential information. A malicious user could intercept the keystrokes and compromise the security of your computer and data.

[Show All](#)

Automate sending a report snapshot in e-mail

1. Create a [macro](#).

[How?](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
 2. In the **Action** column, click the arrow to display the [action list](#).
 3. Click the action you want to use.
 4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
 5. Type a comment for the action. Comments are optional.
 4. To add more actions to the macro, move to another [action row](#) and repeat step 3. Microsoft Access carries out the actions in the order you list them.
2. Add the **SendObject** action to the macro. For example:

Argument	Value
Object Type	Report
Object Name	Summary of Sales by Quarter
Output Format	Snapshot Format

To All Northwind Traders; Shareholders
Cc none
Bcc none
Subject Quarterly Sales report
Message Text This is our best quarter in years!
Edit Message No

3. Run the macro each time you want to send the [report snapshot](#).

[How?](#)

To run a macro directly, do one of the following:

- To run a macro from the [Macro window](#), click **Run**  on the [toolbar](#).
- To run a macro from the [Database window](#), click **Macros**  and then double-click a macro name.
- On the **Tools** menu, point to **Macro**, click **Run Macro**, and then select the macro from the **Macro Name** list. Then click a macro in the **Macro Name** box.
- Run a macro from a [Microsoft Visual Basic](#) procedure by using the **RunMacro** method of the **DoCmd** object.

[Show All](#)

Troubleshoot report snapshots

[I can't open a report snapshot.](#)

- You might not have permission to open the file or the folder that contains the file. Check the [permissions](#) on the file or folder, or contact your system administrator.
- The report snapshot is too big, and you don't have enough disk space. Reclaim more disk space so that the complete report snapshot can be saved to your disk.
- The file is not a report snapshot. A snapshot file has a file extension of .snp and must be in snapshot format. Verify that the file has the correct extension and is in report snapshot format (in Microsoft Windows Explorer, right-click the file and click **Properties** on the shortcut menu; the type should be Snapshot File).
- You don't have [Snapshot Viewer](#) installed. If necessary, Snapshot Viewer is automatically installed by Microsoft Access the first time you create a report snapshot, but you might not have created the report snapshot. You can also install Snapshot Viewer from the Setup program, or from a download page located on the Microsoft Office Developer's Web site.
- If you are attempting to open the report snapshot from a Web page, you might experience a connection or timeout problem while the report snapshot is being downloaded. Redisplay the Web page, or restart your Web browser to see whether that fixes the problem.

[I get the Web page instead of the report when I print a report snapshot.](#)

- If you are viewing a report snapshot from a Web page, don't use the **Print** command on the **File** menu. That **Print** command prints only the contents of the Web page, not each page of the report snapshot. Instead, use the **Print** button  to the right of the navigation controls at the bottom of the window, or right-click the report snapshot, and then click **Print** on the shortcut menu.

[I can't send a report snapshot in e-mail.](#)

- If the **Mail Recipient (as Attachment)** subcommand on the **Send To** command of the **File** menu isn't available, your mail program might not be properly installed, or it might not support Messaging Application Programming Interface (MAPI). Check your program's documentation to confirm that it supports MAPI, and if so, try to reinstall the mail program to make the **Mail Recipient (as Attachment)** subcommand available.
- Depending on how your e-mail program was set up, you might not be able to send a report snapshot programmatically if the e-mail program requires you to log on and enter information. For example, Microsoft Outlook can be installed to prompt you for a user profile when you log on. Therefore, you might need to modify your e-mail program.

About updating data

[When can I update data from a query? \(MDB\)](#)

[About updating data in an Access project](#)

[Show All](#)

Set the default view of a database object

1. Open a table, query, or form in [Design view](#).
2. Do one of the following:
 - On a table or query, click **Properties**  on the toolbar.
 - On a form, double-click the [form selector](#).
3. Set the **DefaultView** property by selecting a value from the list.

Note If you want to disable a view for a form, you can set one or more of the following properties: **AllowFormView**, **AllowDatasheetView**, **AllowPivotTableView**, and **AllowPivotChartView**.

Stop a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

- Press CTRL+BREAK.

[Show All](#)

Run a query (MDB)

Do one of the following:

[Run a select or crosstab query](#)

When you open a [select](#) or [crosstab query](#), Microsoft Access runs (executes) the query for you and shows the results in [Datasheet view](#).

1. In the [Database window](#), click **Queries**  under **Objects**.
2. Click the query you want to open.
3. Click **Open** on the Database window toolbar.

[Run an action query](#)

Unlike [select](#) and [crosstab queries](#), you can't view the results of an [action query](#) by opening it in [Datasheet view](#). However, in Datasheet view you can preview the data that will be affected when you run the action query.

Caution It's a good idea to make a copy of the data you are changing or moving in an action query, in case you need to restore the data to its original state after running the action query.

1. Open the action query in [Design view](#).
2. To preview the records that will be affected in Datasheet view, click **View**  on the toolbar and check the records. For each action query, you will see the following:

For this query	The datasheet displays
Update	The fields to be updated.
Delete	The records to be deleted.
Make-table	The fields to be included in the new table.
Append	The records to be added to another table.

3. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
4. Click **Run**  on the toolbar to run the query.

Note To stop a query after you start it, press CTRL+BREAK.

[Show All](#)

When can I update data from a query? (MDB)

In some cases, you can edit data in query [Datasheet view](#) to change the data in the underlying table. In other cases, you can't. The following information shows whether a query's results can be updated, and if not, whether there is an alternative.

[Data is updatable](#)

You can update a query or query field in the following cases:

- A query based on one table
- A query based on tables with a [one-to-one relationship](#)
- The query's results contain a [Memo](#), [Hyperlink](#), or [OLE Object](#)

[Data is updatable under certain conditions](#)

If a query is based on tables with a [one-to-many relationship](#), you might not be able to edit the data for the following query fields.

Query field	Solution
Join field from the "one" side	Enable cascading updates between the two tables.
New records, if the "many" side join field doesn't appear in the datasheet	Add the join field from the "many" side to your query to allow adding new records.
Join field from the "many" side, after you've updated data on the "one" side	Save the record; then you'll be able to make changes to the "many" side join field.
Blank field from the table on the "one" side of a one-to-many relationship where an outer join exists	Enter values in fields from the table on the "many" side, but only if the joined field from the "one" side contains a value for that record.

New records, if entire unique key of [ODBC](#) table isn't output Select all [primary key](#) fields of ODBC tables to allow inserts into them.

[Data can be deleted but not updated](#)

Query or query field	Solution
Query (or underlying table) for which Update Data permission isn't granted	To modify data, permissions must be assigned.
Query (or underlying table) for which Delete Data permission isn't granted	To delete data, permissions must be assigned.

[Data can't be updated](#)

Query or query field	Solution
Query based on three or more tables in which there is a many-to-one-to-many relationship	Though you can't update the data in the query directly, you can update the data in a form or data access page based on the query if the form's RecordsetType property is set to Dynaset (Inconsistent Updates) .
Crosstab query	None
SQL pass-through query	None
Query that calculates a sum, average, count or other type of total on the values in a field, or an update query that references a field in the Update To row from either a crosstab query, select query , or subquery that contains totals or aggregate functions	By using a domain aggregate function in the Update To row of an update query, you can reference fields from either a crosstab query, select query, or subquery that contains totals or aggregate functions.
Union query	None
Query whose UniqueValues property is set to Yes	None
Query that includes a linked ODBC table with no unique index , or a Paradox table without a primary key	None

Query that includes more than one table or query, and the tables or queries aren't joined by a join line in [Design view](#)

You must join the tables properly in order to update them.

[Calculated field](#)

None

Field is read-only; the database was opened as read-only or is located on a read-only drive

None

Field in record that has been deleted or [locked](#) by another user

A locked record should be updatable as soon as it is unlocked.

[Show All](#)

Troubleshoot queries (MDB)

All queries

[I'm getting an unexpected message.](#)

[I'm getting an SQL syntax message.](#)

- You may have typed incorrect syntax in [SQL view](#).
- You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS [New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created

under one ANSI SQL query mode to an Access database set to another mode.

- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

 [I'm getting the message "Can't bind name <name>."](#)

You may encounter this message if you didn't explicitly declare in the **Query Parameters** dialog box the parameters used for a [crosstab query](#) or a query that a crosstab query or chart is based on. To resolve this, do the following:

1. In the query that contains the parameter, specify the parameter and its data type in the **Query Parameters** dialog box.

 [How?](#)

1. On the **Query** menu, click **Parameters**.
2. In the first **Parameter** cell, type the first prompt you entered in the query [design grid](#).
3. In the **Data Type** cell to the right, click the appropriate [data type](#) according to the following guidelines:

If the parameter

field data type is **Click this data type in the Data Type cell**

Currency,

Date/Time, Memo, Currency, Date/Time, Memo, OLE Object, Text, OLE Object, Text, and Yes/No and Yes/No

Number

Byte, Single, Double, Integer, Long Integer, and Replication ID. These data types correspond to the five **FieldSize** property settings of the Number data type.

Unknown

Value. This is a generic data type that accepts any type of data.

Binary

Binary. Used with parameter queries directed to linked tables that do recognize this data type (Microsoft Access does not recognize this data type).

4. Repeat Steps 2 and 3 for each parameter you want to specify a data type for.

2. In the same query, set the **ColumnHeadings** property.

[How?](#)

You can change the order or limit the column headings in a [crosstab query](#). For example, in column headings containing the months of the year, you can display the months chronologically rather than alphabetically. Or, you can limit the columns to just January through June.

1. Open the crosstab query in [Design view](#).
2. Click the background of query Design view, outside the [design grid](#) and the [field lists](#).
3. On the toolbar, click **Properties**  to display the query's property sheet.
4. In the **ColumnHeadings** property box, enter the column headings you want to display, in the order in which you want to display them. Between the column headings, type a comma or the list separator for your country/region. (To find the list separator for your country/region,

see the regional settings in Microsoft Windows Control Panel.)

The column headings you enter must exactly match the column headings in the query datasheet. For example, if a column heading in the datasheet is "USA," you must enter a column heading of "USA"—not "US." (After you press ENTER or move the pointer to a different location, Microsoft Access places quotation marks around each heading.)

5. To view the query's results, click **View**  on the toolbar.

Notes

- If you run a crosstab query often, or if you use it as the basis for a form, report, or [data access page](#), you can speed up the query by using the preceding procedure to specify fixed column headings.
- If you frequently use the same column headings in different queries, consider creating a table with one [Text](#) field to store the column headings. Then open the table and copy the headings into the **ColumnHeadings** property box as needed.

[I'm getting a key violation message.](#)

A key violation occurs if:

- You attempt to run a query that appends, deletes, or updates records in a way that violates the rules of [referential integrity](#) for related tables.
- Your query attempts to append or update records that contain [primary key](#) values that already exist in the destination table.

If you run such a query, it will not modify the records that cause the key violation.

To modify records in a way that will violate referential integrity, you can break the relationship between the affected tables or turn referential integrity off, and then modify the affected tables one at a time. However, the resulting data in the two tables may then conflict.

[I'm getting a validation rule violation message.](#)

A [validation rule](#) violation occurs if you attempt to update or append records that violate the validation rules for a field or record.

[Restrict or validate data](#)

A field validation rule is used to check the value entered into a field as the user leaves the field. A record validation rule controls when an entire record can be saved. Unlike field validation rules, record validation rules can refer to other fields.

You can also determine whether data that already existed before you specified a validation rule or changed the **Required** or **AllowZeroLength** property violates the current settings.

1. Open a table in [Design view](#).
2. Do one of the following:

[Define a validation rule to control what values can be entered into a field](#)

1. In the upper portion of the window, click the field you want to define a validation rule for.
2. In the lower portion of the window, click the **ValidationRule** property box, and then type the validation rule, or click the **Build** button  to create the validation rule using the [Expression Builder](#).

For example, you could define the validation expression ">9" for a Quantity field to prevent a user from placing an order for fewer than 10 units.

3. In the **ValidationText** property box, type the message that you want displayed when the rule is broken.

For example, for the validation expression ">9", you could enter "You must order 10 or more units."

4. If you set a validation rule in a field that contains data, Microsoft Access will ask if you want to apply the new rule to existing data when you save the table. If you click **Yes**, Access will warn you when

existing data violates the validation rule.

[Define a validation rule to control when a record can be saved](#)

1. Click **Properties**  on the toolbar to display the table's property sheet.
2. In the **ValidationRule** property box, type the validation rule. Or click the **Build** button  to create the validation rule using the [Expression Builder](#).

For example, you could define the validation expression "[RequiredDate]<=[OrderDate]+30" to make sure that the date entered into the RequiredDate field is within 30 days of the date in the OrderDate field.

3. In the **ValidationText** property box, type the message that you want Microsoft Access to display when the rule is broken.

For example, for the validation expression "[RequiredDate]<=[OrderDate]+30", you could enter "The required date must be within 30 days of the order date."

4. If you set a validation rule in a table that contains data, Microsoft Access will ask if you want to apply the new rule to existing data when you save the table. If you click **Yes**, Microsoft Access will warn you when existing data violates the validation rule.

[Test data that already existed before you specified a validation rule](#)

- o Right click the title bar of the table, and click **Test Validation Rules**.

Microsoft Access will warn you if existing data violates a validation rule or the **Required** or **AllowZeroLength** settings.

[I'm getting the message "Query contains ambiguous outer joins."](#)

When you see the following message:

"The SQL statement couldn't be executed, because it contains ambiguous outer

joins."

You tried to execute an SQL statement that contains multiple [joins](#). There are three types of joins in Microsoft Access: [inner joins](#) (also called equi-joins), left [outer joins](#), and right outer joins. In some multiple joins, the results of the query can differ depending on the type of join and the order in which the joins are performed.



[Example of a multiple join using three tables](#)

For example, if you have three tables, Consultants, Active Consultants, and Projects, and you create an SQL query by joining these tables in the following way:

```
Consultants LEFT OUTER JOIN Active Consultants INNER JOIN Projects
```

Microsoft Access can process this query in two ways:

- (Consultants LEFT OUTER JOIN Active Consultants) INNER JOIN Proj

Access first creates a left outer join between the Consultants and Active Consultants tables, and then creates an inner join between the results of those two tables and the Projects table.

- Consultants LEFT OUTER JOIN (Active Consultants INNER JOIN Proje

Access creates a left outer join between the Consultants table and the results of the inner join between the Active Consultants and Projects tables

Because Microsoft Access can produce different results between these two ways of joining, you must specify which way to join by changing one of the joins or by separating the query into two queries.



[Multiple joins that aren't ambiguous](#)

The following combinations of joins can only produce one result and will not generate an "ambiguous outer joins" message:

- Consultants INNER JOIN Active Consultants INNER JOIN Projects
- Consultants INNER JOIN Active Consultants LEFT OUTER JOIN Projects
- Consultants LEFT OUTER JOIN Active Consultants LEFT OUTER JOIN Projects
- Consultants RIGHT OUTER JOIN Active Consultants INNER JOIN Projects
- Consultants RIGHT OUTER JOIN Active Consultants LEFT OUTER JOIN Projects
- Consultants RIGHT OUTER JOIN Active Consultants RIGHT OUTER JOIN Projects

[Multiple joins that are ambiguous](#)

The following combinations of joins can produce more than one result and will generate an "ambiguous outer joins" message:

- Consultants INNER JOIN Active Consultants RIGHT OUTER JOIN Projects
- Consultants LEFT OUTER JOIN Active Consultants INNER JOIN Projects
- Consultants LEFT OUTER JOIN Active Consultants RIGHT OUTER JOIN Projects

[I'm getting a type mismatch message.](#)

- Verify that the [criteria](#) you specified is for the same [data type](#) as the data in the underlying table or query. For example, the field ReorderLevel has a Number data type. Therefore, if you type the criteria "50", you'll get an error because Microsoft Access interprets values in quotation marks as text, not numbers.

Other situations that cause a data type conflict include:

- You're specifying criteria for a [Lookup field](#) and the criteria uses the

values that display on the lookup list instead of their associated [foreign key](#) values. Because the foreign key values are the values actually stored in the underlying table, you should use them when you specify criteria for a field.

- You typed the dollar sign (\$) in criteria you specified for a Currency field. Remove the dollar sign, and then view the results.
- Verify that the data type of each pair of [joined](#) fields in the query is the same. If not, change the data type of one of the joined fields to match the data type of the other.

 [I'm getting the message "Too few parameters expected."](#)

You may encounter this message if a [parameter query](#) exported to the [IDC](#) file type doesn't have data types specified in the **Query Parameters** dialog box.

1. In the [Database window](#), click **Queries**  under **Objects**.
2. Click the name of the parameter query you want to modify, and click **Design** on the Database window toolbar.
3. Specify a data type for the parameter.

 [How?](#)

1. On the **Query** menu, click **Parameters**.
2. In the first **Parameter** cell, type the first prompt you entered in the query [design grid](#).
3. In the **Data Type** cell to the right, click the appropriate [data type](#) according to the following guidelines:

If the parameter field data type is

Currency,

Date/Time, Memo, Currency, Date/Time, Memo, OLE Object, Text, OLE Object, Text, and Yes/No

and Yes/No

Number

Click this data type in the Data Type cell

Byte, Single, Double, Integer, Long Integer, and Replication ID. These data types correspond to

the five **FieldSize** property settings of the Number data type.

Unknown

Value. This is a generic data type that accepts any type of data.

Binary

Binary. Used with parameter queries directed to linked tables that do recognize this data type (Microsoft Access does not recognize this data type).

4. Repeat Steps 2 and 3 for each parameter you want to specify a data type for.

4. Save and run the query.

[I'm getting an error message about s_Generation in my append query.](#)

You can get this message in a [replicated](#) database when you run an [append query](#) that contains the asterisk (*) in the **Field** row in the [design grid](#). The asterisk selects all the fields from one of the underlying tables. To avoid this problem, remove the asterisk from the design grid, and then add each field from the table's [field list](#) to the design grid by double-clicking the title bar of the field list.

[I'm getting the message "Operation must use an updatable query."](#)

This error occurs when the current query's **Update To** row includes a field from either a crosstab query or select query in which an [aggregate](#) (total) was calculated for the field (using either the **Totals** row or a [domain function](#) in the **Field** row). To update a field using the aggregate of another field, calculate the aggregate in the update query itself, not a different query.

1. In query [Design view](#) for the current query, remove the field you added from the crosstab or select query (where an aggregate was calculated for the field).
2. In the **Update To** cell of the field you want to update, enter a domain function that calculates the same aggregate as is calculated for that field in the select or crosstab query.

For example, if you had a [calculated field](#) named Sales So Far in your Products table, you could update it with the product of the Quantity and

UnitPrice fields in the Order Details table by adding it to the design grid in the update query and then entering the following domain function in its **Update To** cell.

```
DSum("[Quantity] * [UnitPrice]", "Order Details", "[ProductID]='
```

3. To run the query, click **Run**  on the toolbar.
4. To see the results, open the table by clicking **Tables**  under **Objects** in the Database window and clicking **Open** on the Database window toolbar.

 [The wildcard character in my query is behaving unexpectedly.](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible and use different wildcard characters. For example, if you run:

- An ANSI-89 SQL [query](#) in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use because if you create a later query under a different ANSI SQL query mode than the current mode of your Access database, your query could produce runtime errors or unexpected results.

 [About using wildcard characters to search for partial or matching values](#)

You use wildcard characters as placeholders for other characters when you are specifying a value you want to find and you:

- Know only part of the value.
- Want to find values that start with a specific letter or match a certain pattern.

Wildcard characters are meant to be used with fields that have the Text data type. You can sometimes use them successfully with other data types, such as dates, if you don't use the Microsoft Windows Control Panel to change the regional settings for these data types.

If you are using Microsoft Jet database engine SQL in a [Microsoft Access database](#), you can use the following wildcard characters in [queries](#) and [expressions](#) to find such things as field values, records, or file names. You can also use these characters in the **Find** and **Replace** dialog boxes in an Access database or a [Microsoft Access project](#).

Character	Description	Example
*	Matches any number of characters. It can be used as the first or last character in the character string.	wh* finds what, white, and why
?	Matches any single alphabetic character.	B?ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill b[!ae]ll finds bill and bull but not ball or bell
!	Matches any character not in the brackets.	
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd
#	Matches any single numeric character.	1#3 finds 103, 113, 123

To find such things as field values, records, or file names in a Microsoft Access project, or in a Microsoft Access database that uses Microsoft SQL Server-compatible syntax, use the following ANSI-92 wildcards.

Note If you are using the **Find** and **Replace** dialog boxes in an Access project or in an Access database that uses Microsoft SQL Server-compatible syntax, you must use the Microsoft Jet SQL wildcards.

Character	Description	Example
%	Matches any number of characters. It can be used as the first or last character in the character string.	wh% finds what, white, and why
_	Matches any single alphabetic character.	B_ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill b[^ae]ll finds bill and bull but not ball or bell
^	Matches any character not in the brackets.	
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd

Notes

- When using wildcard characters to search for other wildcard characters such as an asterisk (*), question mark (?), number sign (#), opening bracket ([), or hyphen (-), you must enclose the item you're searching for in brackets. If you're searching for an exclamation point (!) or a closing bracket (]), you don't need to enclose it in brackets.

For example, to search for a question mark, type **[?]** in the **Find** dialog box. If you're searching for a hyphen and other characters simultaneously, place the hyphen before or after all the other characters inside the brackets. (However, if you have an exclamation point (!) after the opening bracket, place the hyphen after the exclamation point.)

- You can't search for the opening and closing brackets ([]) together because Microsoft Access interprets this combination as a [zero-length string](#). You must enclose the opening and closing brackets in brackets ([[]]).

[The same underlying query behaves differently in a data access page and](#)

[form.](#)

The [query](#) underlying a [form](#) runs under the current [ANSI SQL query mode](#) setting of the [Microsoft Access database](#). On the other hand, the query underlying a [data access page](#) always runs under ANSI-92 SQL query mode because a page connects to your Access database by using [ADO](#) and the Microsoft Jet OLE DB provider, both of which can only use ANSI-92 SQL. Therefore, if the SQL query mode setting of your Access database is ANSI-89, the query underlying the form runs as an ANSI-89 query, but the query underlying a page always runs under ANSI-92 regardless of the Access database setting. To resolve this, use a different query for the form and, if possible, rewrite the query for the form to produce the same results as the underlying query for the page.

[I'm not getting the records I want.](#)

[There are too many records.](#)

- If tables or queries in a query aren't [joined](#) to one another, either directly or indirectly, Microsoft Access doesn't know which records are associated with which, so it displays every combination of records (called a "cross-product" or "Cartesian product") between the two tables. Therefore, if each table had 10 records in it, the query's results will contain 100 records (10 X 10). It also means the query might take a long time to run and ultimately might produce meaningless results. To prevent this, create a join between the tables or queries.
- If tables or queries are already joined, check the join properties of each join line to see what type of join is in effect. Join properties determine whether one table or query contributes all of its records or only those records that match records in the other table.
- Add [criteria](#) to limit the records that the query retrieves. Only records that meet the criteria you define are retrieved.
- If you already specified criteria, you may want to specify additional criteria, either for the same field using the **And** operator, or for other fields. Also, verify that existing criteria is entered correctly. Check for misspellings, unnecessary spaces, or extra characters. Verify that you set the criteria for the appropriate field and that it's the criteria you want the records to meet. If

you're using alternative criteria in the **Or** row in the query [design grid](#) or in criteria [expressions](#) that use the **Or** operator, you may want to remove it.

- Check the data you're trying to find. If the data contains extra spaces or other special characters, the criteria must account for this. You can use wildcard characters in the criteria if you aren't sure of or don't want to specify the exact field values. If you're already using wildcard characters and not getting the records you want, you may be using the wildcard characters on the wrong data type.
- In a [select](#), [append](#), or [make-table query](#), if you're using **Top Values**  on the toolbar in query [Design view](#) (or setting the **TopValues** property in the query's property sheet) to return a specified number of the highest or lowest values in a field, the query might return more records than the number you specified. This is because all records that match the last record are also returned. Therefore, a query would return all three of the following records even if you requested to see the employees with the top two sales totals because the query doesn't distinguish between the two 50,000 sales totals.

EmployeeID Sales

1	60,000
2	50,000
3	50,000

If you don't want duplicate values to be returned, set the **UniqueValues** property in the query property sheet to **Yes**.

 [There are too few records.](#)

- If the query includes more than one table or query, check the [join](#) properties of each join line between them. Join properties determine if a table or query contributes all of its records or only those records that match records in the other table.
- If the type of join you're using retrieves only matching records from the two joined tables, any records in which the joined field contains a **Null** value won't be included in the query's results. You can use the **Nz** function to convert **Null** values to a zero, zero-length string, or other specified value so that those records are included in the query's results.
- If you specified [criteria](#), it may be too restrictive. For example, if you

specified criteria in the **Criteria** row for two different fields, both criteria must be true for a record before the query retrieves it. You may want to delete some or all of the criteria to allow the query to return more records.

- If you have entered criteria that combine two or more [expressions](#) with an **OR** operator on more than one criteria row, you must repeat any other expressions in other fields for each criteria row you have used in order for those other expressions to apply to all criteria. For example, if your combined criteria for a ShipCountry field is "Canada" OR "UK", and you want to limit the ExtendedPrice field to <10000 for both countries, enter all the criteria on one criteria row, or repeat the <10000 expression for each criteria row you use.
- Verify that existing criteria is entered correctly. Check for misspellings, unnecessary spaces, or extra characters. Verify that you set the criteria for the appropriate field and that it's the criteria you want the records to meet.
- If a field name includes a colon, you must enclose the name in brackets whenever you refer to the field in an expression. Otherwise, Microsoft Access interprets the field name as a literal text value, and therefore won't retrieve the records you want.
- Use wildcard characters in the criteria if you aren't sure of or don't want to specify the exact field values. If you're already using wildcard characters and not getting the records you want, you may be using the wildcard characters on the wrong data type.
- Add alternative criteria to select the records you want to work with. You can do this in the query [design grid](#) in the **Or** row, or in the **Criteria** row using expressions containing the **Or** operator.
- If you're using criteria to select records containing specific values in a field or fields, any records with **Null** values in that field don't match the value, and therefore won't be included in the records that are returned. To include records containing **Null** values, type **Is Null** in the **Or** row.

[The records are wrong.](#)

- Check the data you're trying to find. If the data contains extra spaces or other special characters, the [criteria](#) must account for this. You can use wildcard characters in the criteria if you aren't sure of or don't want to specify the exact field values. If you're already using wildcard characters and not getting the records you want, you may be using the wildcard characters on the wrong data type.
- Verify that existing criteria is entered correctly. Check for misspellings,

unnecessary spaces, or extra characters. Verify that you set the criteria for the appropriate field and that it's the criteria you want the records to meet.

- Verify that the tables and queries you've added to the upper part of query [Design view](#) are the correct ones.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/region beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by

helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[I'm not getting the columns I want.](#)

[There are too many columns.](#)

- Remove the fields in the query [design grid](#) that you don't want to see in the query results.
- Remove the asterisk (*) if you added it to the design grid. The asterisk automatically includes all fields from the underlying table. If you want to keep the asterisk (*), clear the **Show** check box for all fields except the asterisk; otherwise those fields will appear twice in the query's results.
- Hide fields that you don't need to display in the query's results.
- Verify that the **Output All Fields** check box isn't selected. (To see if it is selected, on the **Tools** menu, click **Options**, and then click the **Tables/Queries** tab.) If it is, all fields from the underlying table or query will appear in the query's results without being added to the query design grid. If you then add a field to the design grid, say, for setting [criteria](#), you need to clear the **Show** check box for that field; otherwise, the column appears twice in the query's results. If you want, clear the **Output All Fields** check box. Changing this option affects the property setting for new queries you create, but it doesn't affect existing queries.

[There are too few columns.](#)

- Add more fields to the query [design grid](#) or, if you add the asterisk (*) to the design grid, the query automatically includes all fields from the underlying table or query.
- Add other tables or queries, then add the other fields you need.
- Show hidden fields.
- If the **Output All Fields** check box is not selected, only the fields in the design grid are retrieved. (To see if it is selected, on the **Tools** menu, click **Options**, and then click the **Tables/Queries** tab.) If it is, all fields from the underlying table or query will appear in the query's results without being

added to the query design grid. If you then add a field to the design grid, say, for setting [criteria](#), you need to clear the **Show** check box for that field; otherwise, the column appears twice in the query's results. Or, if you want, clear the **Output All Fields** check box. Changing this option only affects the property setting for new queries you create, not existing queries.

[The columns are wrong.](#)

- In the query [design grid](#), remove the fields that you don't want the query to retrieve.
- If columns are missing, add their fields to the query design grid.
- Verify that the tables and queries you're retrieving records from are the correct ones.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is

not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.

- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[I'm not getting the calculation results I expected.](#)

[The results of a totals \(aggregate\) calculation are wrong.](#)

- If there are records with **Null** (blank) values in the field you are calculating, those records won't be included in the calculation.
- If the query isn't performing the calculation on the correct groups, verify that you have **Group By** in the **Total** cell for the field or fields you want to use for grouping calculations. Also, where you specify [criteria](#) in the query design grid determines if records are excluded before the query groups or performs calculations, if groups are excluded before the calculation is performed, or if certain results are excluded after the calculation is performed.
- In a [crosstab query](#), you might have too many or too few **Group By** fields. Add, remove, or change the fields designated as row headings that group the totals vertically. Verify that you're using the correct field's values as column headings that group the totals horizontally.
- Verify that the [aggregate](#) you selected is the correct one. If not, select a different aggregate from the list in the **Total** cell.

[The results of a totals \(aggregate\) calculation are in the wrong column.](#)

In the query [design grid](#), delete the [aggregate](#) you specified in the **Total** cell for that field (column), and then select one in a different field.

[The arithmetic calculation in the design grid returns a Null \(blank\) value.](#)

If you use an arithmetic operator (such as +, -, *, /) in an [expression](#), such as [UnitsInStock] + [UnitsOnOrder], and one of the fields in the expression contains a **Null** value, then the result of the entire expression will be a **Null** value. To avoid this, you can convert **Null** values to zeros or other numbers by using the **Nz** function.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 -2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in

the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

Specific types of queries

[Select queries](#)

[My AutoLookup query isn't filling in data.](#)

For AutoLookup to work, certain conditions must be met:

- The query must be based on more than one table and the tables must have a [one-to-many relationship](#). ([Referential integrity](#) doesn't have to be enforced.)
- The [join](#) field on the "one" side of the relationship must have a [unique index](#). A unique index means that the field is a [primary key](#) or its **Indexed** property in table [Design view](#) is set to **Yes (No Duplicates)**.
- The join field you add to the [design grid](#) must come from the table on the "many" side of the one-to-many relationship. (In other words, it's the [foreign key](#) for that table.) For AutoLookup to work, this field can't be a primary key and its **Indexed** property can't be set to **Yes (No Duplicates)**. For example, in a query that includes the Customers and Orders tables, drag to the design grid the CustomerID field from the Orders table, not the one from the Customers table.
- The value you enter in the join field from the "many" side must already exist in the join field from the "one" side.

[Microsoft Access automatically joins tables or queries I don't want joined.](#)

Even if you haven't created [relationships](#) between tables, Microsoft Access automatically creates joins if you add two tables to a query and the tables each have a field with the same or compatible [data type](#) and if one of the join fields is a [primary key](#). You can enable or disable these automatic joins.

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab.
3. Select or clear the **Enable AutoJoin** check box.

Note This setting applies to new queries only.

[I can't update data from a query.](#)

In some cases, you can edit data in query [Datasheet view](#) to change the data in the underlying table. In other cases, you can't. The following information shows whether a query's results can be updated, and if not, whether there is an alternative.

[Data is updatable](#)

You can update a query or query field in the following cases:

- A query based on one table
- A query based on tables with a [one-to-one relationship](#)
- The query's results contain a [Memo](#), [Hyperlink](#), or [OLE Object](#)

[Data is updatable under certain conditions](#)

If a query is based on tables with a [one-to-many relationship](#), you might not be able to edit the data for the following query fields.

Query field	Solution
Join field from the "one" side	Enable cascading updates between the two tables.
New records, if the "many" side join field doesn't appear in the datasheet	Add the join field from the "many" side to your query to allow adding new records.
Join field from the "many" side, after you've updated data on the "one" side	Save the record; then you'll be able to make changes to the "many" side join field.
Blank field from the table on the "one" side of a one-to-many relationship where an outer join exists	Enter values in fields from the table on the "many" side, but only if the joined field from the "one" side contains a value for that record.
New records, if entire unique key of ODBC table isn't output	Select all primary key fields of ODBC tables to allow inserts into them.

[Data can be deleted but not updated](#)

Query or query field	Solution
Query (or underlying table) for which Update Data permission isn't granted	To modify data, permissions must be assigned.
Query (or underlying table) for which Delete Data permission isn't granted	To delete data, permissions must be assigned.

[Data can't be updated](#)

Query or query field	Solution
Query based on three or more tables in which there is a many-to-one-to-many relationship	Though you can't update the data in the query directly, you can update the data in a form or data access page based on the query if the form's RecordsetType property is set to Dynaset (Inconsistent Updates) .
Crosstab query	None
SQL pass-through query	None
Query that calculates a sum, average, count or other type of total on the values in a field, or an update query that references a field in the Update To row from either a crosstab query, select query , or subquery that contains totals or aggregate functions	By using a domain aggregate function in the Update To row of an update query, you can reference fields from either a crosstab query, select query, or subquery that contains totals or aggregate functions.
Union query	None
Query whose UniqueValues property is set to Yes	None
Query that includes a linked ODBC table with no unique index , or a Paradox table without a primary key	None
Query that includes more than one table or query, and the tables or queries aren't joined by a join line in Design view	You must join the tables properly in order to update them.
Calculated field	None
Field is read-only; the database was	

opened as read-only or is located on a read-only drive None

Field in record that has been deleted or [locked](#) by another user A locked record should be updatable as soon as it is unlocked.

[I get prompted for a parameter when I don't expect it.](#)

If you get prompted to type parameters when you attempt to run a query, report, or [data access page](#), and you don't expect this prompt, one of the following might apply:

- You deleted a parameter from the query [design grid](#) but not from the **Query Parameters** dialog box.
- You renamed or misspelled a field in the query, report, or data access page but not in the query's underlying table(s).
- You have a field that refers to a [calculated field](#). If a field in the query performs a calculation based on the calculated value, make sure the **Show** check box is selected for the calculated field in the query design grid.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've

created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.

- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[Parameter queries](#)

[I get prompted for a parameter when I don't expect it.](#)

If you get prompted to type parameters when you attempt to run a query, report, or [data access page](#), and you don't expect this prompt, one of the following might apply:

- You deleted a parameter from the query [design grid](#) but not from the **Query Parameters** dialog box.
- You renamed or misspelled a field in the query, report, or data access page but not in the query's underlying table(s).
- You have a field that refers to a [calculated field](#). If a field in the query performs a calculation based on the calculated value, make sure the **Show** check box is selected for the calculated field in the query design grid.

[I want the option of returning all records with a parameter query.](#)

1. In the [design grid](#) under the field that contains the prompt in the **Criteria** cell, type [*prompt*] **Is Null** in the **Or** cell, where *prompt* is the same prompt that is in the **Criteria** cell for that field. For example:

In this cell	Type
Criteria	Like [Enter Category ID:]
Or	[Enter Category ID:] Is Null

2. Run the query.
3. Leave the **Enter Parameter Value** dialog box blank, when Microsoft Access prompts you.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[Crosstab queries](#)

[The column headings aren't in the order I want, or I want to exclude some of them.](#)

You can set the query's **ColumnHeadings** property to specify which column headings are displayed and in what order you want them to appear.

[I can't update the data in my crosstab query.](#)

Data in a [crosstab query](#) is not updatable.

[One or more column headings in the query's results are labeled "<>".](#)

Microsoft Access returns "<>" as the column name for any **NullNull** value in the field with **Column Heading** in its **Crosstab** cell. To avoid this you can:

- Explicitly define your column headings by setting the query's **ColumnHeadings** property.

[Sort or limit column headings displayed in a crosstab query](#)

You can change the order or limit the column headings in a [crosstab query](#). For example, in column headings containing the months of the year, you can display the months chronologically rather than alphabetically. Or, you can limit the columns to just January through June.

1. Open the crosstab query in [Design view](#).
2. Click the background of query Design view, outside the [design grid](#) and the [field lists](#).
3. On the toolbar, click **Properties**  to display the query's property sheet.
4. In the **ColumnHeadings** property box, enter the column headings you want to display, in the order in which you want to display them. Between the column headings, type a comma or the list separator for your country/region. (To find the list separator for your country/region, see the regional settings in Microsoft Windows Control Panel.)

The column headings you enter must exactly match the column headings in the query datasheet. For example, if a column heading in the datasheet is "USA," you must enter a column heading of "USA"— not "US." (After you press ENTER or move the pointer to a different location, Microsoft Access places quotation marks around each heading.)

5. To view the query's results, click **View**  on the toolbar.

Notes

- If you run a crosstab query often, or if you use it as the basis for a form, report, or [data access page](#), you can speed up the query by using the preceding procedure to specify fixed column headings.
- If you frequently use the same column headings in different queries, consider creating a table with one [Text](#) field to store the column headings. Then open the table and copy the headings into the **ColumnHeadings** property box as needed.
- In the [design grid](#), type **Is Not Null** in the **Criteria** cell of the designated **Column Heading** field.
- In the **Criteria** cell of the designated **Column Heading** field, use the Nz function in an [expression](#) that will convert **Null** values to a more meaningful value, such as "Unknown."

 [The wrong field values are showing as row or column headings.](#)

Verify that the field whose values you want to use as column headings is in the query [design grid](#) with **Column Heading** in the **Crosstab** cell, and that the field

or fields whose values you want to use as row headings are in the query design grid with **Row Heading** in the **Crosstab** cell. To change the value in a **Crosstab** cell, click the cell, and then click a value from the list. If you want to display only certain values as column headings, set the query's **ColumnHeadings** property.

 [Sort or limit column headings displayed in a crosstab query](#)

You can change the order or limit the column headings in a [crosstab query](#). For example, in column headings containing the months of the year, you can display the months chronologically rather than alphabetically. Or, you can limit the columns to just January through June.

1. Open the crosstab query in [Design view](#).
2. Click the background of query Design view, outside the [design grid](#) and the [field lists](#).
3. On the toolbar, click **Properties**  to display the query's property sheet.
4. In the **ColumnHeadings** property box, enter the column headings you want to display, in the order in which you want to display them. Between the column headings, type a comma or the list separator for your country/region. (To find the list separator for your country/region, see the regional settings in Microsoft Windows Control Panel.)

The column headings you enter must exactly match the column headings in the query datasheet. For example, if a column heading in the datasheet is "USA," you must enter a column heading of "USA"—not "US." (After you press ENTER or move the pointer to a different location, Microsoft Access places quotation marks around each heading.)

5. To view the query's results, click **View**  on the toolbar.

Notes

- If you run a crosstab query often, or if you use it as the basis for a form, report, or [data access page](#), you can speed up the query by using the preceding procedure to specify fixed column headings.
- If you frequently use the same column headings in different queries, consider creating a table with one [Text](#) field to store the column headings. Then open the table and copy the headings into the **ColumnHeadings**

property box as needed.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[Action queries \(append, update, delete\)](#)

[The append query appended data to or from the wrong fields.](#)

Verify that you have selected the correct fields for the data you want to append, entered the correct [criteria](#) in the query [design grid](#), and selected the correct fields to which you want to append data. Restore the appended query to its original state, through a backup copy if one is available, then do one or more of the following:

[Select the fields whose data you want to append](#)

Add the following fields to the query design grid for your append query:

- The fields you want to append.
- The fields you want to specify criteria for.
- The field corresponding to the [primary key](#) field of the table you're appending records to.

[Select the fields to which you want to append data](#)

- In the **Append To** cell of the fields whose data you want to append, click the name of the field you want to add the data to.

Note If you added the asterisk (*) in the **Field** cell, select the asterisk in the **Append To** cell. All the field names in both tables must be exactly the same. You can't specify criteria for selecting records in the column that contains the asterisk, but you can specify criteria in a separate column.

[The update query didn't perform the update I wanted.](#)

- You need to specify a different update by changing the [expression](#) in the **Update To** cell. For example, to raise salaries by 5 percent, type **[Salary] * 1.05** in the **Update To** cell of the Salary field.
- Verify that you're updating data in the correct field. If you're updating data

in the wrong field, specify an update for a different field. To specify an update for a field, type the expression you want in the **Update To** cell.

[I don't know the best way to delete duplicate records after using the Find Duplicates Wizard.](#)

Delete the duplicate record or the original directly in query [Datasheet view](#).

[Delete a record](#)

[Delete a record in a datasheet or form](#)

1. Open a [datasheet](#), or open a form in [Form view](#).
2. Click the record you want to delete.
3. Click **Delete Record**  on the toolbar.

Note When you delete data, you might want to delete related data in other tables. For example, if you delete a supplier, you probably want to delete the products that the supplier supplies. In some cases, you can make sure the proper data is deleted by enforcing [referential integrity](#) and turning on cascade deletions.

[Delete a record in a data access page](#)

Important Once you delete a record, you cannot undo the deletion from the database.

- On the record navigation toolbar, click **Delete** .

Note If you don't see the **Delete** button, it means one of the following:

- The page does not support deletions.
- The designer of the page has customized the button. Look for custom instructions on how to use the page, or contact the designer.

If you want Microsoft Access to delete all the duplicates and keep the original records automatically, don't use the Find Duplicates Wizard or convert a Find Duplicates query to a delete query.

[Delete duplicate records from a table](#)

Deleting duplicate records from a table requires two steps. First, you create a copy of the structure of the table that contains duplicates, and then make [primary keys](#) of all the fields that contain duplicates. Second, you create and run an [append query](#) from the original table to the new table. Because fields that are primary keys can't contain duplicate records, this procedure produces a table without duplicate records.

1. Copy the table and make primary keys of fields with duplicates.

[How?](#)

1. In the [Database window](#), click **Tables**  under **Objects**.
 2. Click the name of the table you want to delete duplicate records from.
 3. Click **Copy**  on the [toolbar](#).
 4. Click **Paste**  on the toolbar.
 5. In the **Paste Table As** dialog box, type a name for the copied table, click **Structure Only**, and then click **OK**.
 6. Open the new table in [Design view](#), and select the fields that contained duplicates in the table you copied.
 7. Click **Primary Key**  on the toolbar to create a primary key based on the selected fields.
 8. Save and close the table.
2. Append only unique records to the new table.

[How?](#)

1. Create a new query based on the original table that contains duplicates.
2. In query [Design view](#), click **Query Type**  on the toolbar, and then click **Append Query**.
3. In the **Append** dialog box, click the name of the new table in the **Table Name** list, and then click **OK**.
4. Include all the fields from the original table by dragging the asterisk (*) to the query [design grid](#).
5. Click **Run**  on the toolbar.
6. Click **Yes** when you receive the message that you're about to append rows.

7. Click **Yes** when you receive the message that Microsoft Access can't append all the records in the append query. This transfers only unique records to your new table and discards the duplicates.
8. Open the table to see the results.
9. When you're sure the new table has the correct unique records, you can delete the original table, and then rename the new table using the name of the original table.

[The delete query deleted records from the wrong table.](#)

Make sure the delete query includes:

- The table you want to delete records from. Drag the asterisk (*) from the [field list](#) for that table to the query [design grid](#).
- The field(s) you want to specify [criteria](#) for. Drag the field you want from the table's field list to the design grid.

In the query design grid, **From** or **Where** is displayed in the **Delete** row. **From** is displayed in the **Delete** cell of the table you'll delete records from, and **Where** is displayed in the **Delete** cell of the field(s) you specify criteria for.

[The query is running under a different ANSI SQL query mode.](#)

You created your [query](#) under a different [ANSI SQL query mode](#) than the current mode of your [Microsoft Access database](#).

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've

created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.

- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and preventing the changing of the query mode through the application's user interface.

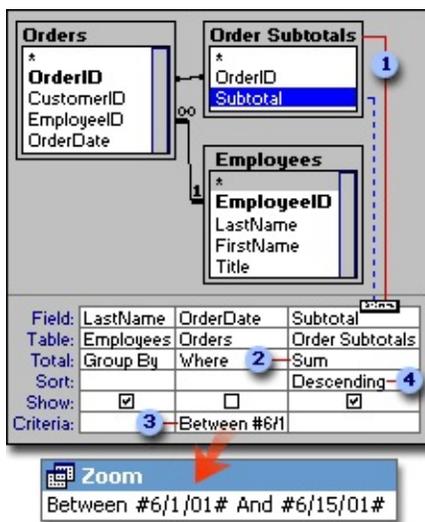
- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[Show All](#)

About designing a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

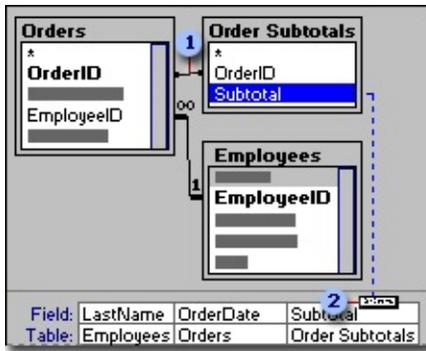
When you open a query in [Design view](#), or open a form, report, or datasheet and show the [Advanced Filter/Sort window](#), you see the [design grid](#), which you can use to make a variety of changes to get the query results you want.



- 1 Add or remove tables, queries, and fields
- 2 Calculate amounts
- 3 Limit results using criteria
- 4 Sort records

[Add or remove tables, queries, and fields](#)

You can add a table or query if the data you need isn't in the query, or remove a table or query if you decide you don't need them. Once you add the tables or queries you need, you can then add the fields that you want to work with to the design grid, or remove them if you decide you don't need them.

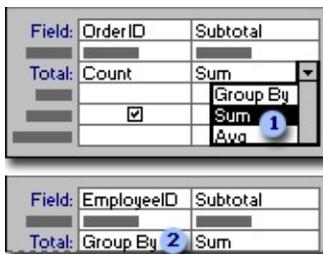


1 A join line between [field lists](#) tells Microsoft Access how the data in one table is related to the data in the other.

2 You drag a field from the field list to a column in the design grid to show the field in the query results.

[Calculate amounts](#)

You can add the values in a field or do other computations with the data by specifying the type of calculation to perform.



1 Use an [aggregate function](#), such as **Sum** or **Avg**, to calculate one amount for all the records in each field in the design grid.

2 Use **Group By** to calculate separate amounts for groups of records in a field.

[Limit results by using criteria](#)

You can limit the records that you see in the query's results or the records that are included in a calculation by specifying criteria.

Field:	LastName	OrderDate
Table:	Employees	Orders
Total:	Group By	Where
Sort:		
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteria:		Between #6/1/01# And #6/15/01#

Criteria:	Between #6/1/01# And #6/15/01#
or:	Between #6/1/01# And #6/15/01#

Field:	LastName	OrderDate	Subtotal
Table:	Employees	Orders	Order Subtotals
Total:	Group By	Where	Sum
Sort:			Descending
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Between #6/1/01# And #6/15/01#	> 100000

- 1 To limit the records in the query's results, enter criteria in one or more fields.
- 2 Use the **Or** row for alternative criteria in the same field.
- 3 Enter criteria for different fields. For example, for orders between 6/1/01 and 6/15/01 ...
- 4 ... calculate total order amounts, but display only those that are more than \$100,000.

[Sort records](#)

You can sort the query's results by specifying a sort order in the design grid.

Field:	LastName	OrderDate	Subtotal
Table:	Employees	Orders	Order Subtotals
Total:	Group By	Where	Sum
Sort:			Descending
Show:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:		Between #6/1/01# And #6/15/01#	> 100000

- 1 If you specify a sort order for more than one field, Microsoft Access sorts the leftmost field first, so you should arrange the fields you want to sort from left to right in the design grid.
- 2 Sort by ascending or descending order, or remove a sort.

[Use the asterisk in the query design grid](#)

To include all fields from a table in a query, you can either select each field individually or use the asterisk (*) wildcard character. Selecting the asterisk has

an advantage over selecting all the fields: When you use the asterisk, the query results automatically include any fields that are added to the underlying table or query after the query is created, and automatically exclude fields that are deleted.

When you use the asterisk, you must add fields along with the asterisk to the [design grid](#) before you can sort records or specify criteria for those fields.

When you drag more than one field at a time, Microsoft Access places each field in a separate column. If you drag the asterisk to the grid, Microsoft Access places the table or query name in one column and attaches a period and asterisk to the name (for example, **Customers.***). After either operation, the datasheet looks the same.

If you type an asterisk in the **Field** row rather than dragging it, you must type the table name also. For example, type **Customers.***.

Field:

[Considerations when designing or modifying queries in a multiuser environment](#)

If you open a query to make design changes and someone else modifies the query design before you have finished, Microsoft Access displays a message telling you that the design has changed since you started working. You can either replace the new changes with your changes or save your version of the query with a different name. If others are using the query to view data while you're changing the design, they don't see the design changes until they close and reopen the query.

[Show All](#)

About types of queries (MDB)

You use queries to view, change, and analyze data in different ways. You can also use them as a source of records for forms, reports, and [data access pages](#). There are several types of queries in Microsoft Access.

[Select queries](#)

A select query is the most common type of query. It retrieves data from one or more tables and displays the results in a datasheet where you can update the records (with some restrictions). You can also use a select query to group records and calculate sums, counts, averages, and other types of totals.

[Parameter queries](#)

A parameter query is a query that when run displays its own dialog box prompting you for information, such as [criteria](#) for retrieving records or a value you want to insert in a field. You can design the query to prompt you for more than one piece of information; for example, you can design it to prompt you for two dates. Access can then retrieve all records that fall between those two dates.

Parameter queries are also handy when used as the basis for forms, reports, and data access pages. For example, you can create a monthly earnings report based on a parameter query. When you print the report, Access displays a dialog box asking for the month that you want the report to cover. You enter a month and Access prints the appropriate report.

[Crosstab queries](#)

You use crosstab queries to calculate and restructure data for easier analysis of your data. Crosstab queries calculate a sum, average, count, or other type of total for data that is grouped by two types of information— one down the left side of the datasheet and another across the top.

[Action queries](#)

An action query is a query that makes changes to or moves many records in just one operation. There are four types of action queries:

- **Delete Queries** A delete query deletes a group of records from one or more tables. For example, you could use a delete query to remove products that are discontinued or for which there are no orders. With delete queries, you always delete entire records, not just selected fields within records.
- **Update Queries** An update query makes global changes to a group of records in one or more tables. For example, you can raise prices by 10 percent for all dairy products, or you can raise salaries by 5 percent for the people within a certain job category. With an update query, you can change data in existing tables.
- **Append Queries** An append query adds a group of records from one or more tables to the end of one or more tables. For example, suppose that you acquire some new customers and a database containing a table of information on those customers. To avoid typing all this information into your own database, you'd like to append it to your Customers table.
- **Make-Table Queries** A make-table query creates a new table from all or part of the data in one or more tables. Make-table queries are helpful for creating a table to export to other [Microsoft Access databases](#) or a history table that contains old records.

[SQL queries](#)

An SQL query is a query you create by using an [SQL statement](#). You can use Structured Query Language (SQL) to query, update, and manage relational databases such as Access.

When you create a query in query [Design view](#), Access constructs the equivalent SQL statements behind the scenes for you. In fact, most query properties in the property sheet in query Design view have equivalent clauses and options available in [SQL view](#). If you want, you can view or edit the SQL statement in SQL view. However, after you make changes to a query in SQL view, the query might not be displayed the way it was previously in Design view.

Some SQL queries, called [SQL-specific queries](#), can't be created in the [design](#)

[grid](#). For [pass-through](#), [data-definition](#), and [union queries](#), you must create the SQL statements directly in SQL view. For [subqueries](#), you enter the SQL in the **Field** row or the **Criteria** row of the query design grid.

[Show All](#)

Show only the high or low values in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

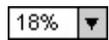
You can show in a query datasheet those records with the highest or lowest values in a field or the highest or lowest percentage of values in the field. For example, you may want to show the top 10 salespeople with the highest total sales for a month, or the bottom 20 percent of students in a class by grade average.

1. Create a query in [Design view](#).

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. Add to the [design grid](#) the fields you want to display in the query's results, including the field you want to display top values for.
3. In the **Sort** cell of the field you want to display top values for, click **Descending** to display the highest values or **Ascending** to display the lowest values. If you're sorting on other fields in the query's design grid as well, those fields must be to the right of the top values field.
4. Click in the **Top Values** box  on the toolbar.
5. Enter the percentage or the number of highest or lowest values you want the query results to display.

Note To display a percentage, enter a number followed by a percent sign (%).



6. To view the query's results, click **View**  on the toolbar.

Note Setting the **TopValues** property in the query's property sheet works the same as using the **Top Values** button on the toolbar.

[Show All](#)

Save the SQL statement for a form or report as a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

By saving as a query the [SQL statement](#) displayed in the **RecordSource** property of a form or report, you can run the query independently, or base new forms and reports on the query.

1. Open a form or report in [Design view](#).
2. Double-click the [form selector](#) or the [report selector](#) to open the property sheet for the form or report.
3. Next to the **RecordSource** property box, click the **Build** button  to display query Design view.
4. Click **Save**  on the toolbar.
5. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules, and then click **OK**.

Microsoft Access adds the query to **Queries**  under **Objects** in the [Database window](#).

[Show All](#)

About select and crosstab queries (MDB)

Select and crosstab queries are the two main ways you query a [Microsoft Access database](#) to retrieve just the data you want.

[Select queries](#)

[What is a select query?](#)

A select query is the most common type of query. You use it to:

- Retrieve data from one or more tables by using [criteria](#) you specify and then display the data in the order you want.
- Update records in the datasheet of a select query (with some restrictions).
- Group records and calculate sums, counts, averages, and other types of totals.

1

Product Name	Supplier	Phone
Alice Mutton	Pavlova, Ltd.	(03) 444-2343
Aniseed Syrup	Exotic Liquids	(71) 555-2222
Camembert Pierrot	Gai pâturage	38.76.98.58

2

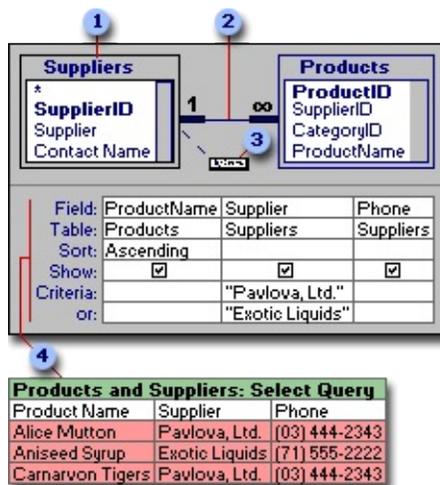
Name	Total Orders	Total Sales
Andrew Fuller	125	\$197,110.84
Anne Dodsworth	53	\$86,737.34

- 1 Bring together data from multiple tables and sort it in a particular order.
- 2 Perform calculations on groups of records.

[Creating a select query](#)

You create a query with a wizard or from scratch in query [Design view](#). In Design view, you specify the data you want to work with by adding the tables or

queries that contain the data, and then by filling in the [design grid](#).



- 1 [Field lists](#) show the fields in the tables or queries you add to your query.
- 2 A join line tells Microsoft Access how data in one table or query is related to data in another table or query.
- 3 You add fields to the design grid by dragging them to the field lists.
- 4 The fields, sort order, and [criteria](#) you add to the design grid determine what you will see in the query's results.

[Crosstab queries](#)

[What is a crosstab query?](#)

You use crosstab queries to calculate and restructure data for easier analysis of your data. Crosstab queries calculate a sum, average, count, or other type of total for data that is grouped by two types of information— one down the left side of the datasheet and another across the top.

1

Last Name	Category Name	Sum of Subtotal
Buchanan	Beverages	\$46,302.09
Buchanan	Condiments	\$16,789.35
Buchanan	Confections	\$36,182.13
Callahan	Beverages	\$111,047.76
Callahan	Condiments	\$49,566.21
Callahan	Confections	\$80,005.35

2

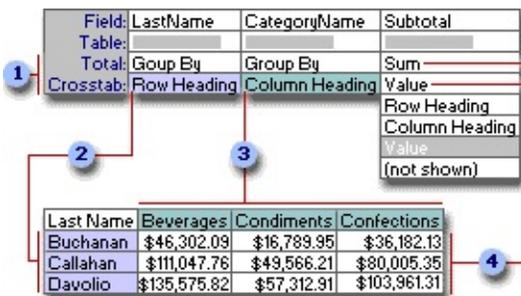
Last Name	Beverages	Condiments	Confections
Buchanan	\$46,302.09	\$16,789.35	\$36,182.13
Callahan	\$111,047.76	\$49,566.21	\$80,005.35

1 This select query only groups the totals vertically by employee and category. This results in more records, making comparisons between different employees' totals more difficult.

2 A crosstab query displays the same information, but groups it both horizontally and vertically so the datasheet is more compact and easier to analyze.

[Creating a crosstab query](#)

You create a [crosstab query](#) with a wizard or from scratch in query [Design view](#). In the [design grid](#), you specify which field's values will become column headings, which field's values will become row headings, and which field's values to sum, average, count, or otherwise calculate.



1 The settings in these rows determine how data is shown.

2 This setting displays the field's values as row headings.

3 This setting displays the field's values as column headings.

4 These settings display the total orders.

[Let Microsoft Access create a select or crosstab query for you](#)

Access can often create a query for you so you don't have to design one from scratch.

- To create a query to use as the basis of a form, report, or [data access page](#), try the form, report, or data access page wizards. They create the form, report or data access page, and if it's based on more than one table, they also create its underlying [SQL statement](#). If you want, you can save the SQL statement as a query.
- To easily create queries that you want to run independently or base multiple forms, reports, and data access pages on, try one of the query wizards. Query wizards do all the basic work for you after you provide answers to a series of questions. Even if you've created many queries, you may want to use a wizard to quickly design the query. Then you can switch to [Design view](#) to customize it.
- To create queries from [filters](#) you created using [Filter By Form](#), [Filter By Selection](#), or [Filter For Input](#), save the filter as a query.

If none of these methods satisfies your needs, you can create the query from scratch in query Design view.

[Show All](#)

Create a select or crosstab query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Create a select query

[Create a simple select query with a wizard](#)

The Simple Select Query Wizard creates queries that retrieve data from the fields you specify in one or more tables or queries. If you want, the wizard can also sum, count, and average values for groups of records or all records, and it can calculate the minimum or maximum value in a field. However, you can't limit the records it retrieves by setting [criteria](#).

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Simple Query Wizard**, and then click **OK**.
3. Follow the directions in the wizard dialog boxes. In the last dialog box, you can choose either to run the query or to see the query's structure in [Design view](#).

If the resulting query isn't exactly what you want, you can rerun the wizard or change the query in Design view.

[Create a select query on my own](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then

click **Close**.

5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.

[Find duplicate records or field values in a table with a wizard](#)

Using the Find Duplicates Query Wizard, you can create a [select query](#) to determine if there are duplicate records in a table. For example, you might search for duplicate values in an Address field to determine if you have duplicate records for the same supplier, or you might search for duplicate values in a City field to see which suppliers are in the same city.

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Find Duplicates Query Wizard**, and then click **OK**.
3. Follow the directions in the wizard dialog boxes. If you don't choose to show fields in addition to those with duplicate values, the query results will sum the instances of each duplicate value. In the last dialog box, you can choose to run the query or see the query's structure in [Design view](#).

If the resulting query isn't exactly what you want, you can rerun the wizard or change the query in Design view.

[Find unmatched records between tables with a wizard](#)

Using the Find Unmatched Query Wizard, you can create a [select query](#) to find records in one table that don't have related records in another table. For example, you can find customers who don't have orders.

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Find Unmatched Query Wizard**, and then click **OK**.
3. Follow the directions in the wizard dialog boxes. In the last dialog box, you can choose to run the query or see the query's structure in [Design view](#).

If the resulting query isn't exactly what you want, you can rerun the wizard

or change the query in Design view.

Create a crosstab query

[Create a crosstab query with a wizard](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Crosstab Query Wizard**, and then click **OK**.
3. Follow the directions in the wizard dialog boxes. In the last dialog box, you can choose to run the query or see the query's structure in [Design view](#).

If the resulting query isn't exactly what you want, you can rerun the wizard or change the query in Design view.

[Create a crosstab query on my own](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#) and specify [criteria](#).
6. On the toolbar, click **Query Type** , and then click **Crosstab**.
7. For the field or fields whose values you want to appear as row headings, click the **Crosstab** row, and then click **Row Heading**.

You must leave the default **Group By** in the **Total** row for these fields.

8. For the field whose values you want to appear as column headings, click the **Crosstab** row, and then click **Column Heading**. You can choose **Column Heading** for one field only, and you must leave **Group By** in the **Total** row for this field.

By default, the column headings are sorted in alphabetic or numeric order. If you want them to appear in a different order, or if you want to limit

which column headings to display, set the query's **ColumnHeadings** property.

[How?](#)

You can change the order or limit the column headings in a [crosstab query](#). For example, in column headings containing the months of the year, you can display the months chronologically rather than alphabetically. Or, you can limit the columns to just January through June.

1. Open the crosstab query in [Design view](#).
2. Click the background of query Design view, outside the [design grid](#) and the [field lists](#).
3. On the toolbar, click **Properties**  to display the query's property sheet.
4. In the **ColumnHeadings** property box, enter the column headings you want to display, in the order in which you want to display them. Between the column headings, type a comma or the list separator for your country/region. (To find the list separator for your country/region, see the regional settings in Microsoft Windows Control Panel.)

The column headings you enter must exactly match the column headings in the query datasheet. For example, if a column heading in the datasheet is "USA," you must enter a column heading of "USA"—not "US." (After you press ENTER or move the pointer to a different location, Microsoft Access places quotation marks around each heading.)

5. To view the query's results, click **View**  on the toolbar.

Notes

- If you run a crosstab query often, or if you use it as the basis for a form, report, or [data access page](#), you can speed up the query by using the preceding procedure to specify fixed column headings.
- If you frequently use the same column headings in different queries, consider creating a table with one [Text](#) field to store the column headings. Then open the table and copy the headings into the **ColumnHeadings** property box as needed.

9. For the field whose values you want to use in the cross-tabulation, click the **Crosstab** row, and then click **Value**.

Only one field can be set to **Value**.

10. In the **Total** row for this field, click the type of [aggregate function](#) you want for the cross-tabulation (such as **Sum**, **Avg**, or **Count**).
11. Do one of the following:

[Specify criteria that limit row headings before the calculation is performed](#)

- o Enter an [expression](#) in the **Criteria** row for a field with **Row Heading** in the **Crosstab** cell.

For example, you might display sales totals for products in certain categories, such as meat and seafood.

[Specify criteria that limit records before the row headings are grouped and before the cross-tabulation is performed](#)

1. Add the field you want to set criteria for to the design grid.
2. Click **Where** in the **Total** cell.
3. Leave the **Crosstab** cell blank.
4. Enter an [expression](#) in the **Criteria** row.

The query results won't display fields that have **Where** in the **Total** row.

To view the query's results, click **View**  on the toolbar.

To stop the query after you start it, press CTRL+BREAK.

Notes

- The query Quarterly Orders by Product in the Northwind sample database is an example of a [crosstab query](#). To view this query, open the Northwind

database in your Office folder's Samples folder, and then open Quarterly Orders by Product in [Design view](#).

- If you include a field in the query design grid, but click the **(Not Shown)** option in the **Crosstab** cell and **Group By** in the **Total** cell, Microsoft Access groups on it as a **Row Heading**, but doesn't display the row in the query's results.
- The values in your **Column Heading** field may include characters usually not allowed in field names, such as decimals. If they do, Access replaces the character with an underscore in the datasheet.

[Show All](#)

Update one table based on another table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

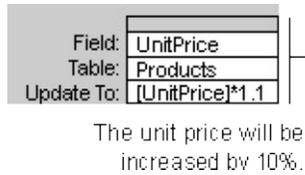
1. Create an [update query](#) that contains the table you want to update and the table whose values you want to copy.

[How?](#)

1. Create a query with the tables or queries that include the records you want to update.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the [toolbar](#), and then click **Update Query**.
3. Drag from the [field list](#) to the query [design grid](#) the fields you want to update or you want to specify [criteria](#) for.
4. In the **Criteria** cell, specify the criteria if necessary.
5. In the **Update To** cell for the fields you want to update, type the [expression](#) or value you want to use to change the fields, as shown in the following illustration.



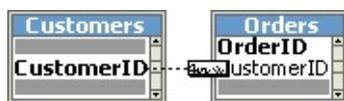
6. To see a list of the records that will be updated, click **View**  on the toolbar. This list won't show the new values. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
 7. Click **Run**  on the toolbar to update the records.
2. If the tables aren't already [joined](#), join them on the fields that have related information.

[How?](#)

In most cases, you want to bring together or perform an action on data from more than one table or query. For example, you might want to view a customer's information with the orders the customer placed. To see this information, you need data from the Customers and Orders tables.

[Join different tables and queries](#)

- o In query [Design view](#), drag a field from the [field list](#) for one table or query to the equivalent field (a field of the same or compatible [data type](#) containing similar data) in the field list for the other table or query.



With this type of [join](#), Microsoft Access selects records from both tables or queries only when the values in the joined fields are equal.

Note Join numeric fields only if the **FieldSize** property settings for both are **Byte**, **Integer**, or **Long Integer**.

In some cases, you want to join two copies of the same table or query, called a [self-join](#), that combines records from the same table when there are

matching values in the joined fields. For example, say you have an Employees table in which the ReportsTo field for each employee's record displays his or her manager's ID instead of name. You could use a self-join to display the manager's name in each employee's record instead.

[Join two copies of the same table or query](#)

1. In query [Design view](#), add the table to the query twice.
2. Create the [join](#) by dragging a field in the first table's [field list](#) to the field you want to relate it to in the second field list.

Note Microsoft Access appends "_1" to the table name in the second field list. For example, if you add the Employees table twice, the first field list is titled "Employees," and the second field list is titled "Employees_1." You can rename the table to something more descriptive by setting the **Alias** or **Caption** property for the field list.

[Example of a self-join using the Employees table](#)

To display the managers' names in the ReportsTo field instead of the managers' IDs, follow these steps.

1. In query Design view, add the Employees table to the query twice.
2. Rename the second table in the query.

[How?](#)

Note Renaming copies of a table or query in a query doesn't rename the underlying table or query.

1. Open a query in [Design view](#).
 2. Click anywhere in the second copy of the [field list](#) for the table or query, and then click **Properties**  on the toolbar to display the property sheet.
 3. In the **Alias** property box, type a new name for the table or query.
3. Create the join by dragging the ReportsTo field in the first table's field list to the EmployeeID field in the second table's field list.

4. Add the LastName, FirstName, and Title fields from the first field list to the [design grid](#).
5. Add the LastName field from the second field list to the design grid. To display the field name as "Manager" instead of "LastName," set the **Caption** property in this field's property sheet.

For example, if you want to copy data from the ProductName field in the Products table to a field in another table, join the two tables on the [primary key](#), which might be called ProductID.

3. In the **Update To** cell for the fields you want to update, type an [expression](#) with the following syntax:

[tablename].[fieldname]

where *tablename* and *fieldname* are the names of the table and field that contain the data you're copying.

For example, if you are copying data from the ProductName field in the Products table, you would type **[Products].[ProductName]** in the **Update To** cell.

Create a table from another table with a query

[Create a table from another table with a query \(MDB\)](#)

[Create a make-table query \(ADP\)](#)

[Show All](#)

Delete duplicate records from a table (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Deleting duplicate records from a table requires two steps. First, you create a copy of the structure of the table that contains duplicates, and then make [primary keys](#) of all the fields that contain duplicates. Second, you create and run an [append query](#) from the original table to the new table. Because fields that are primary keys can't contain duplicate records, this procedure produces a table without duplicate records.

1. Copy the table and make primary keys of fields with duplicates.

 [How?](#)

1. In the [Database window](#), click **Tables**  under **Objects**.
 2. Click the name of the table you want to delete duplicate records from.
 3. Click **Copy**  on the [toolbar](#).
 4. Click **Paste**  on the toolbar.
 5. In the **Paste Table As** dialog box, type a name for the copied table, click **Structure Only**, and then click **OK**.
 6. Open the new table in [Design view](#), and select the fields that contained duplicates in the table you copied.
 7. Click **Primary Key**  on the toolbar to create a primary key based on the selected fields.
 8. Save and close the table.
2. Append only unique records to the new table.

 [How?](#)

1. Create a new query based on the original table that contains duplicates.
2. In query [Design view](#), click **Query Type**  on the toolbar, and then click **Append Query**.

3. In the **Append** dialog box, click the name of the new table in the **Table Name** list, and then click **OK**.
4. Include all the fields from the original table by dragging the asterisk (*) to the query [design grid](#).
5. Click **Run**  on the toolbar.
6. Click **Yes** when you receive the message that you're about to append rows.
7. Click **Yes** when you receive the message that Microsoft Access can't append all the records in the append query. This transfers only unique records to your new table and discards the duplicates.
8. Open the table to see the results.
9. When you're sure the new table has the correct unique records, you can delete the original table, and then rename the new table using the name of the original table.

[Show All](#)

About action queries (MDB)

An action query is a query that makes changes to or moves many records in just one operation. There are four types of action queries: delete, update, append, and make-table.

[Delete query](#)

A delete query deletes a group of records from one or more tables. For example, you could use a delete query to remove products that are discontinued or for which there are no orders. With delete queries, you always delete entire records, not just selected fields within records.

You can use a delete query to delete records from a single table, from multiple tables in a [one-to-one relationship](#), or from multiple tables in a [one-to-many relationship](#), if [cascading deletes](#) are enabled. For example, you could use a delete query to delete all customers from Ireland and all their orders. However, if you need to include the "many" table along with the "one" table in order to add criteria, you must run the query twice, since a query cannot delete records from the primary table and the related tables at the same time.

You can distinguish a one-to-many relationship in query [Design view](#) by looking at the [joins](#) between tables. If one end of a join is marked with the infinity symbol , it is a one-to-many relationship. If both ends are marked with a **1**, it's a one-to-one relationship.

[Important considerations when using a delete query](#)

- Once you delete records using a [delete query](#), you can't undo the operation. Therefore, you should preview the data that the query selected for deletion before you run the query. You can do this by clicking **View**  on the toolbar, and viewing the query in [Datasheet view](#).
- You should maintain backup copies of your data at all times. If you delete the wrong records, you can retrieve them from your backup copies.
- In some cases, running a delete query might delete records in related tables, even if they're not included in the query. This can happen when your query

contains only the table that's on the "one" side of a [one-to-many relationship](#), and you've enabled [cascading deletes](#) for that relationship. When you delete records from the "one" table, you'll also delete records from the "many" table.

- When a delete query contains more than one table, such as a query that deletes duplicate records from one of the tables, the query's **UniqueRecords** property must be set to **Yes**.

[Update query](#)

An update query makes global changes to a group of records in one or more tables. For example, you can raise prices by 10 percent for all dairy products, or you can raise salaries by 5 percent for the people within a certain job category. With an update query, you can change data in existing tables.

[Append query](#)

An append query adds a group of records from one or more tables to the end of one or more tables. For example, suppose that you acquire some new customers and a database containing a table of information on those customers. To avoid typing all this information in, you'd like to append it to your Customers table. Append queries are also helpful for:

- Appending fields based on criteria. For example, you might want to append only the names and addresses of customers with outstanding orders.
- Appending records when some of the fields in one table don't exist in the other table. For example, in the Northwind sample database, the Customers table has 11 fields. Suppose that you want to append records from another table that has fields that match 9 of the 11 fields in the Customers table. An append query will append the data in the matching fields and ignore the others.

[Make-table query](#)

A make-table query creates a new table from all or part of the data in one or more tables. Make-table queries are helpful for:

- Creating a table to export to other [Microsoft Access databases](#). For example, you might want to create a table that contains several fields from

your Employees table, and then export that table to a database used by your personnel department.

- Creating forms, reports, or [data access pages](#) that display data from a specified point in time. For example, suppose you want to display a data access page on 15-May-96 that displays the first quarter's sales totals based on the data that was in the underlying tables as of 9:00 A.M. on 1-Apr-96. A data access page based on a query or [SQL statement](#) extracts the most up-to-date data from the tables (the data as of 15-May-96), rather than the records as of a specific date and time. To preserve the data exactly as it was at 9:00 A.M. on 1-Apr-96, create a make-table query at that point in time to retrieve the records you need and store them in a new table. Then use this table, rather than a query, as the basis for the data access page.
- Making a backup copy of a table automatically by using a macro or code.
- Creating a history table that contains old records. For example, you could create a table that stores all your old orders before deleting them from your current Orders table.
- Improving performance of forms, reports, and data access pages based on multiple-table queries or SQL statements. For example, suppose you want to print multiple reports that are based on a five-table query that includes totals. You might be able to speed things up by first creating a make-table query that retrieves the records you need and stores them in one table. Then you can base the reports on this table or specify the table in an SQL statement as the [record source](#) for a form, report, or page, so you don't have to rerun the query each time you open the form, report, or page. However, the data in the table is frozen at the time you run the make-table query.

[Show All](#)

Create a delete query (MDB)

[Important considerations when using a delete query](#)

- Once you delete records using a [delete query](#), you can't undo the operation. Therefore, you should preview the data that the query selected for deletion before you run the query. You can do this by clicking **View**  on the toolbar, and viewing the query in [Datasheet view](#).
- You should maintain backup copies of your data at all times. If you delete the wrong records, you can retrieve them from your backup copies.
- In some cases, running a delete query might delete records in related tables, even if they're not included in the query. This can happen when your query contains only the table that's on the "one" side of a [one-to-many relationship](#), and you've enabled [cascading deletes](#) for that relationship. When you delete records from the "one" table, you'll also delete records from the "many" table.
- When a delete query contains more than one table, such as a query that deletes duplicate records from one of the tables, the query's **UniqueRecords** property must be set to **Yes**.

[Delete records by using a query that includes only the "one" table in a one-to-many relationship](#)

1. Create a new query that contains the "one" table from which you want to delete records.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify

[criteria](#) and a sort order.

6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the toolbar, and then click **Delete Query**.
3. Drag the asterisk (*) from the [field list](#) for the table to the query [design grid](#).

From appears in the **Delete** cell under this field, as shown in the illustration that follows.

4. To specify [criteria](#) for deleting records, drag to the design grid the fields on which you want to set criteria.

Where appears in the **Delete** cell under this field, as shown in the following illustration.

Field:	Orders.*	OrderID
Table:	Orders	Orders
Delete:	From	Where
Criteria:		<10050

Records will be deleted from the Orders table if the Order ID is less than 10050.

5. In the **Criteria** cell for the fields that you have dragged to the grid, type the criteria.
6. To preview the records that will be deleted, click **View**  on the toolbar. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
7. Click **Run**  on the toolbar to delete the records.

[Delete records by using a query that includes both tables in a one-to-many relationship](#)

1. Create a new query that contains the tables from which you want to delete records.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.

3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
 4. Double-click the name of each object you want to add to the query, and then click **Close**.
 5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
 6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the toolbar, and then click **Delete Query**.
 3. From the table with related records (on the "many" side of the [one-to-many relationship](#)), drag the asterisk (*) from the [field list](#) to the query [design grid](#).

From appears in the **Delete** cell under this field, as shown in the illustration that follows.

4. To specify [criteria](#) for deleting records, drag the fields on which you want to set criteria from the primary table (on the "one" side of the one-to-many relationship) to the design grid.

Where appears in the **Delete** cell under this field, as shown in the following illustration.

Field:	Orders.*	Customer ID
Table:	Orders	Customers
Delete:	From	Where
Criteria:		=0047

Records will be deleted from the Orders table if their related Customer ID in the Customers table is 0047.

5. In the **Criteria** cell for the fields that you have dragged to the grid, type the criteria.
6. To preview the records that will be deleted, click **View**  on the toolbar. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
7. Click **Run**  on the toolbar to delete the records from the table on the "many" side.
8. Select the field list for each table on the "many" side and press DELETE to remove it from the query.

9. With just the primary table remaining in the query and the field on which you want to set criteria in the design grid, click **Run**  again.

Microsoft Access deletes the specified records from the "one" table.

[Delete records from a single table or multiple tables that have a one-to-one relationship](#)

1. Create a new query that contains the tables from which you want to delete records.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the toolbar, and then click **Delete Query**.
3. For the tables you want to delete records from, drag the asterisk (*) from the [field list](#) to the query [design grid](#).

From appears in the **Delete** cell under these fields, as shown in the illustration that follows.

4. To specify [criteria](#) for deleting records, drag to the design grid the fields on which you want to set criteria.

Where appears in the **Delete** cell under these fields, as shown in the following illustration.

Field:	Orders.*	OrderID
Table:	Orders	Orders
Delete:	From	Where
Criteria:		<10050

Records will be deleted from the Orders table if the Order ID is less than 10050.

5. In the **Criteria** cell for the fields that you have dragged to the grid, type the criteria.
6. To preview the records that will be deleted, click **View**  on the toolbar. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
7. Click **Run**  on the toolbar to delete the records.

[Show All](#)

Create a table from another table with a query (MDB)

1. Create a query, selecting the tables or queries that contain the records you want to put in the new table.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the [toolbar](#), and then click **Make Table**. The **Make Table** dialog box appears.
3. In the **Table Name** box, enter the name of the table you want to create or replace.
4. Do one of the following:

Click **Current Database**.

Click **Another Database**, and then either type the path of the database where you want to put the new table or click **Browse** to locate the database.

5. Click **OK**.
6. Drag from the [field list](#) to the query [design grid](#) the fields you want in the new table.
7. In the **Criteria** cell for the fields that you've dragged to the grid, type the [criteria](#).
8. To preview the new table before you create it, click **View**  on the toolbar.

To return to query Design view and make changes or run the query, click **View**  on the toolbar again.

9. To create the new table, click **Run**  on the toolbar.

Note The data in the new table you create does not inherit the field properties or the [primary key](#) setting from the original table.

[Show All](#)

Create an append query (MDB)

1. Create a query that contains the table whose records you want to append to another table.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the toolbar, and then click **Append**. The **Append** dialog box appears.
3. In the **Table Name** box, enter the name of the table you want to append records to.
4. Do one of the following:

If the table is in the currently open database, click **Current Database**.

If the table is not in the currently open database, click **Another Database** and type the path of the database where the table is stored or click **Browse** to locate the database. You can also specify a path to a Microsoft FoxPro, Paradox, or dBASE database, or a connection string to an [SQL database](#).

5. Click **OK**.
6. Drag from the [field list](#) to the query [design grid](#) the fields you want to append and any fields you want to use for setting criteria.

If all the fields in both tables have the same names, you can just drag the asterisk (*) to the query design grid. However, if you're working in a

database [replica](#), you'll need to add all the fields instead.

7. If you have a field with an AutoNumber [data type](#), do one of the following:

[Add AutoNumber values automatically](#)

To have Microsoft Access add AutoNumber values automatically, don't drag the AutoNumber field to the query design grid when you create the query.

With this method, Access appends records and automatically inserts AutoNumber values. The first record appended has a value that is one larger than the largest entry that was ever entered in the AutoNumber field (even if the record that contained the largest AutoNumber value has been deleted).

Use this method if the AutoNumber field in the table you're appending to is a [primary key](#), and the original table and the table you're appending to contain duplicate AutoNumber values.

[Keep the AutoNumber values from the original table](#)

To keep the AutoNumber values from the original table, drag its AutoNumber field to the query design grid when you create the query.

8. If the fields you've selected have the same name in both tables, Microsoft Access automatically fills the matching name in the **Append To** row. If the fields in the two tables don't have the same name, in the **Append To** row, enter the names of the fields in the table you're appending to.
9. In the **Criteria** cell for the fields that you have dragged to the grid, type the criteria on which additions will be made.
10. To preview the records that the query will append, click **View**  on the toolbar. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
11. Click **Run**  on the toolbar to add the records.

[Show All](#)

Create an update query (MDB)

1. Create a query with the tables or queries that include the records you want to update.

 [How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the [toolbar](#), and then click **Update Query**.
3. Drag from the [field list](#) to the query [design grid](#) the fields you want to update or you want to specify [criteria](#) for.
4. In the **Criteria** cell, specify the criteria if necessary.
5. In the **Update To** cell for the fields you want to update, type the [expression](#) or value you want to use to change the fields, as shown in the following illustration.

Field:	UnitPrice
Table:	Products
Update To:	[UnitPrice]*1.1

The unit price will be increased by 10%.

6. To see a list of the records that will be updated, click **View**  on the toolbar. This list won't show the new values. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
7. Click **Run**  on the toolbar to update the records.

[Show All](#)

About parameter queries that prompt for criteria (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

A parameter query is a query that, when run, displays its own dialog box prompting you for information, such as criteria for retrieving records or a value you want to insert in a field. You can design the query to prompt you for more than one piece of information; for example, you can design it to prompt you for two dates. Microsoft Access can then retrieve all records that fall between those two dates.

Parameter queries are also handy when used as the basis for forms, reports, and [data access pages](#). For example, you can create a monthly earnings report based on a parameter query. When you print the report, Access displays a dialog box asking for the month that you want the report to cover. You enter a month and Access prints the appropriate report.

You can also do the following with parameter queries, forms, and reports:

- Create a custom form or dialog box that prompts for a query's parameters, instead of using the parameter query's dialog box, and then display the results in a datasheet. For example, you can add buttons and other [controls](#) to a custom dialog box.
- Print the criteria entered in a parameter query for a report in the [report header](#), so you can determine from the report the values used to create it.
- As an alternative to a parameter query, use a form to enter report criteria, and then show the results in a report.

[Show All](#)

Create a parameter query (MDB)

[Create a parameter query](#)

1. Create a select or [crosstab query](#).
2. In query [Design view](#), drag the fields from the [field list](#) to the query [design grid](#).
3. Do one or more of the following:

[Use one parameter](#)

In the **Criteria** cell for the field you want to use as a parameter, type an expression with a prompt enclosed in square brackets. For example, in a field that displays the current number of units in stock, enter the following:

```
<[Number of Units in Stock:]
```

[Use two or more parameters](#)

In the **Criteria** cell for each field you want to use as a parameter, type an expression with prompts enclosed in square brackets.

For example, in a field that displays dates, you can display the prompts "Type the beginning date:" and "Type the ending date:" to specify a range of values:

```
Between [Type the beginning date:] And [Type the ending date:]
```

[Use parameters with wildcards](#)

In the **Criteria** cell for each field you want to use as a parameter, type an expression with a prompt enclosed in square brackets. To prompt the user for one or more characters to search for, and then find records that begin with or contain the characters the user specifies, create a parameter query that uses the LIKE operator and the wildcard symbol (*).

For example, the following statement searches for words that begin with a specified letter:

```
LIKE [Enter the first character to search by: ] & "*"
```

The following statement searches for words that contain the specified character:

```
LIKE "*" & [Enter any character to search by: ] & "*"
```

4. Specify a [data type](#) for the parameters if your parameter query:
 - Is a crosstab query or the underlying query for a crosstab query. (In these cases, you must also set the **ColumnHeadings** property in the crosstab query.)
 - Is the underlying query for a chart.
 - Prompts for a field with the Yes/No data type.
 - Prompts for fields that come from a table in an external [SQL database](#).

[How?](#)

1. On the **Query** menu, click **Parameters**.
2. In the first **Parameter** cell, type the first prompt you entered in the query [design grid](#).
3. In the **Data Type** cell to the right, click the appropriate [data type](#) according to the following guidelines:

If the parameter field data type is

Currency,

Date/Time, Memo, Currency, Date/Time, Memo, OLE Object, Text, OLE Object, Text, and Yes/No and Yes/No

Number

Click this data type in the Data Type cell

Byte, Single, Double, Integer, Long Integer, and Replication ID. These data types correspond to the five **FieldSize** property settings of the Number

Unknown	data type. Value. This is a generic data type that accepts any type of data.
Binary	Binary. Used with parameter queries directed to linked tables that do recognize this data type (Microsoft Access does not recognize this data type).

- Repeat Steps 2 and 3 for each parameter you want to specify a data type for.

To view the results, click **View**  on the toolbar, and then type a value for the parameter.

 [Create a custom dialog box to prompt for your parameter query's criteria](#)

- Create a custom dialog box form.

 [How?](#)

Before you do the following procedure, create the form or report you want.

- Open the form or report in [Design view](#).
- Double-click the [form selector](#) or [report selector](#) to open the [property sheet](#).
- In the **Pop Up** property box, click **Yes**.
- Do one of the following:

 [For a modeless pop-up form or report](#)

- If you want to be able to size the form or report, in the **BorderStyle** property box, click **Thin**.
- Create a macro or an event procedure to open the pop-up form or report.

Note The Suppliers form in Northwind sample application has an example of an event procedure that displays a modeless pop-up

form. Open the Northwind database in the Samples subfolder of your Office folder, and then open the Suppliers form in Design view. Click the **Build** button  next to the **OnClick** property box of the Review Products button.

[For a modal pop-up form or report](#)

1. In the **Modal** property box, click **Yes**.
2. In the **BorderStyle** property box, click **Dialog**.
3. Create a macro or an event procedure to open the pop-up form or report.
5. Attach the macro or an event procedure to a form or report by specifying the macro name or event procedure as the setting for the appropriate event property. For example, type the name of the macro or event procedure in the **OnClick** property box of a command button.

To position a pop-up form or report on the screen, open it in Design view, maximize the Microsoft Access window, position the window where you want it, and then save the form or report. The pop-up window will appear in this location when the form or report is opened (assuming the **AutoCenter** property is set to **No** and you haven't specified its position in the event procedure that opens the pop-up object).

Note If you want to use a form as a normal form as well as a dialog box, you can use the Dialog setting of the OpenForm action to open the form temporarily as a dialog box instead of using this procedure.

2. Create or open the parameter query that will collect its criteria values from the dialog box.
3. Add to the query [design grid](#) the fields associated with each of the dialog box [controls](#) where you'll enter criteria values.
4. For each of these fields, enter an expression in the **Criteria** cell that tells Microsoft Access to refer to the control on the dialog box for the criteria values.
5. Add to the query design grid the fields whose values you want returned.

[Customer Information Example](#)

For example, you could create a dialog box form named Customer Info with a [control](#) for the CustomerID field. By entering a customer ID in the dialog box, you could have the query return the correct customer's company name, address, and city. To create this query, you'd do the following:

1. Add the CustomerID field to the query design grid.
2. In its **Criteria** cell for the CustomerID field, enter the expression that tells Access to refer to the Customer ID control in the dialog box for the criteria values the query uses. In this case, you'd enter the expression **Forms! [Customer Info]! [CustomerID]**.
3. Add to the query design grid the fields whose values you want returned based on the CustomerID value: CompanyName, Address, and City fields.

[Show All](#)

Change a column's width in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#), or open a form or datasheet and show the [Advanced Filter/Sort window](#).
2. Move the pointer to the right edge of the [column selector](#) for the column you want to change, until the pointer turns into a two-way arrow.



- 1** The pointer changes to a two-way arrow.

3. Do one of the following:

To make the column narrower, drag the edge to the left.

To make the column wider, drag the edge to the right.

To adjust the width to the longest visible entry in the [design grid](#), double-click the edge.

Notes

- To adjust the width of several columns at once, select the columns by dragging across their column selectors, and then use the preceding procedure.
- Adjusting column widths in the design grid has no effect on the column width in the datasheet.

[Show All](#)

Add a column in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#), or open a form or datasheet and show the [Advanced Filter/Sort window](#).
2. Click anywhere in the column to the left of which you want to add a new column.
3. On the **Insert** menu, click **Columns**.

[Show All](#)

Add all fields in a table to a query in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To include all fields from a table in a query, you can either select each field individually or use the asterisk (*) wildcard character. Selecting the asterisk has an advantage over selecting all the fields: When you use the asterisk, the query results automatically include any fields that are added to the underlying table or query after the query is created, and automatically exclude fields that are deleted.

When you use the asterisk, you must add fields along with the asterisk to the [design grid](#) before you can sort records or specify criteria for those fields.

When you drag more than one field at a time, Microsoft Access places each field in a separate column. If you drag the asterisk to the grid, Microsoft Access places the table or query name in one column and attaches a period and asterisk to the name (for example, `Categories.*`). After either operation, the datasheet looks the same.

If you type an asterisk in the **Field** row rather than dragging it, you must type the table name also. For example, type **Customers.***.

Field:

[Show All](#)

Add or remove a table or query in the query design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Add a table or query to a query](#)

1. Open a query in [Design view](#).
2. On the toolbar, click **Show Table** .
3. In the **Show Table** dialog box, click the tab that lists the objects whose data you want to work with.
4. Click the name of the object you want to add to the query. To select additional objects one at a time, hold down CTRL while you click each object name. To select a block of objects, click the first name in the block, hold down SHIFT, and then click the last name in the block.
5. Click **Add**, and then click **Close**.

[Tip](#)

You can also add a table or query to a query by dragging the table or query name from the [Database window](#) to the upper part of query Design view.

[Remove a table or query from a query](#)

1. Open a query in [Design view](#).
2. In the upper part of query Design view, select the table or query you want to remove by clicking anywhere in its [field list](#), and then press the DELETE key.

Fields that you have dragged from the field list to the [design grid](#) are also removed from the query. The table or query is not deleted from the database, however.

[Show All](#)

Show or hide table names in the query design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When you add fields from multiple tables to the query [design grid](#), you might want to track which table each field came from by displaying the table names under the field names.

1. Open the query in [Design view](#).
2. On the **View** menu, click **Table Names**.

[Show All](#)

Customize the table and query names in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).



[Rename a table or query that's added to a query](#)

Note Renaming copies of a table or query in a query doesn't rename the underlying table or query.

1. Open a query in [Design view](#).
2. Click anywhere in the second copy of the [field list](#) for the table or query, and then click **Properties**  on the toolbar to display the property sheet.
3. In the **Alias** property box, type a new name for the table or query.



[Show or hide table names in the query design grid by default](#)

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab.
3. Select or clear the **Show Table Names** check box.

Note This setting applies to new queries only.

[Show All](#)

Prevent showing duplicate records in a query (MDB)

Microsoft Access considers a record to be unique when a [value](#) in any field in a record differs from the value in the same field in any other record. In a query, you aren't necessarily displaying all the fields that make up the records in the underlying tables or queries. Therefore, if the field that distinguishes one record from another isn't in the query [design grid](#), the query's results can appear to include duplicate records.

For example, if one of the underlying tables for a query is an Employee table, and you add only the LastName and Country fields to the query design grid, it might appear that you have duplicate records when several employees have the same last name and live in the same country/region. However, the records are not duplicates in the underlying table because the EmployeeID is unique for each record.

1. Open a query in [Design view](#).
2. Select the query by clicking anywhere in query Design view outside the design grid and the [field lists](#).
3. Click **Properties**  on the toolbar to display the query's property sheet.
4. Do one of the following:

Prevent showing duplicate records in a query based on fields in the underlying table or query

- Set the **UniqueRecords** property to **Yes**.

Prevent showing duplicate records in a query based on fields in the query design grid

- Set the **UniqueValues** property to **Yes**.

Notes

- The **UniqueRecords** property has an effect only when you use more

than one table in the query and select fields from those tables. When the **UniqueRecords** property is set to **Yes**, Microsoft Access automatically sets the **UniqueValues** property to **No**.

- When the **UniqueValues** property is set to **Yes**, Microsoft Access automatically sets the **UniqueRecords** property to **No**.

5. To view the query's results, click **View**  on the toolbar.

[Show All](#)

About using criteria to retrieve certain records (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Criteria are limits you place on a query or an advanced filter to identify the specific records you want to work with. For example, instead of viewing all the suppliers that your company uses, you can view just suppliers from Japan. To do this, you specify criteria that limits the results to records whose Country field is "Japan".

To specify criteria for a field in the [design grid](#), enter an [expression](#) in the **Criteria** cell for that field. The expression in the preceding example would be "Japan". You can use more complicated expressions, however, such as "Between 1000 And 5000".

If your query includes linked tables, the values you specify in criteria on fields from the linked tables are [case-sensitive](#)— they must match the case of the values in the underlying table.



[Combining criteria with the And or the Or operator](#)

You can enter additional criteria for the same field or different fields. When you type expressions in more than one **Criteria** cell, Microsoft Access combines them using either the **And** or the **Or operator**. If the expressions are in different cells in the same row, Access uses the **And** operator, which means only the records that meet the criteria in all the cells will be returned. If the expressions are in different rows of the design grid, Access uses the **Or** operator, which means records that meet criteria in any of the cells will be returned.

The following examples show different ways to combine criteria with the **And** operator or the **Or** operator.



[One field using the Or operator](#)

Field:	Company Name	Region
Sort:		
Show:		
Criteria:		"SP"
or:		"RJ"

Query1: Select Query	
Company Name	Region
Comércio Mineiro	SP
Familia Arquibaldo	SP
Gourmet Lanchonetes	SP
Hanari Carnes	RJ
Que Delícia	RJ

- 1 Region is SP or RJ.
- 2 Access shows all company names that meet one of the criteria.

[One field using the And operator](#)

Field:	Company Name	Region
Sort:		
Show:		
Criteria:	Like "The*" And Like "*Box*"	

Query2: Select Query	
Company Name	Region
The Cracker Box	MT

- 1 Company name begins with "The" and includes the word "Box."
- 2 Access shows the company name that meets both criteria.

[Two fields using the Or operator](#)

Field:	Company Name	Region
Sort:		
Show:		
Criteria:	Like "H*"	
or:		"RJ"

Query3: Select Query	
Company Name	Region
Hanari Carnes	RJ
HILARIÓN-Abastos	Táchira
Hungry Coyote Import	OR
Hungry Owl All-Night	Co. Cork
Que Delícia	RJ

- 1 Company name begins with H or region is RJ.
- 2 Access shows all company names that meet the criteria for the first field or the

second field.

[Two fields using the And operator](#)

Field:	Company Name	Region
Sort:		
Show:		
Criteria:	Like "H*"	"RJ"

1

Query4: Select Query	
Company Name	Region
Hanari Carnes	RJ

2

- 1 Company name begins with H and region is RJ.
- 2 Access shows all company names that meet the criteria for the first field and the second field.

[Three fields using the Or operator and the And operator](#)

Field:	Company Name	Region	City
Sort:			
Show:			
Criteria:	Like "H*"	"RJ"	
or:		"SP"	Like "C*"

1

2

Query5: Select Query		
Company Name	Region	City
Gourmet Lanchonetes	SP	Campinas
Hanari Carnes	RJ	Rio de Janeiro

3

- 1 Company name begins with H and region is RJ ...
- 2 ... or region is SP and city begins with C.
- 3 Access shows all company names that meet the criteria for the first or second row of the design grid.

[Where you specify criteria affects when calculations are performed](#)

You can also specify criteria to affect the calculations and produce different query results. By adding criteria, you can limit the:

- Groups before performing calculations on those groups.
- Results after calculations on the groups are performed.
- Records before they are grouped and before calculations are performed.

[Show All](#)

About using wildcard characters

You use wildcard characters as placeholders for other characters when you are specifying a value you want to find and you:

- Know only part of the value.
- Want to find values that start with a specific letter or match a certain pattern.

Wildcard characters are meant to be used with fields that have the Text data type. You can sometimes use them successfully with other data types, such as dates, if you don't use the Microsoft Windows Control Panel to change the regional settings for these data types.

If you are using Microsoft Jet database engine SQL in a [Microsoft Access database](#), you can use the following wildcard characters in [queries](#) and [expressions](#) to find such things as field values, records, or file names. You can also use these characters in the **Find** and **Replace** dialog boxes in an Access database or a [Microsoft Access project](#).

Character	Description	Example
*	Matches any number of characters. It can be used as the first or last character in the character string.	wh* finds what, white, and why
?	Matches any single alphabetic character.	B?ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill b[!ae]ll finds bill and bull but not ball or bell
!	Matches any character not in the brackets.	
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd
#	Matches any single numeric character.	1#3 finds 103, 113, 123

To find such things as field values, records, or file names in a Microsoft Access project, or in a Microsoft Access database that uses Microsoft SQL Server-compatible syntax, use the following ANSI-92 wildcards.

Note If you are using the **Find** and **Replace** dialog boxes in an Access project or in an Access database that uses Microsoft SQL Server-compatible syntax, you must use the Microsoft Jet SQL wildcards.

Character	Description	Example
%	Matches any number of characters. It can be used as the first or last character in the character string.	wh% finds what, white, and why
_	Matches any single alphabetic character.	B_ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill
^	Matches any character not in the brackets.	b[^ae]ll finds bill and bull but not ball or bell
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd

Notes

- When using wildcard characters to search for other wildcard characters such as an asterisk (*), question mark (?), number sign (#), opening bracket ([), or hyphen (-), you must enclose the item you're searching for in brackets. If you're searching for an exclamation point (!) or a closing bracket (]), you don't need to enclose it in brackets.

For example, to search for a question mark, type **[?]** in the **Find** dialog box. If you're searching for a hyphen and other characters simultaneously, place the hyphen before or after all the other characters inside the brackets. (However, if you have an exclamation point (!) after the opening bracket, place the hyphen after the exclamation point.)

- You can't search for the opening and closing brackets ([]) together because Microsoft Access interprets this combination as a [zero-length string](#). You must enclose the opening and closing brackets in brackets ([[]]).

[Show All](#)

About working with blank fields in queries (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When a field contains no values, it contains a **Null** value or, for [Text](#), [Memo](#), or [Hyperlink](#) fields, a **Null** value or a [zero-length string](#). If **Null** values exist in a field, they can affect the query's results. Following are some guidelines for working with **Null** values and zero-length strings in queries.

[Joining tables whose matching fields contain Null values](#)

When you join tables in a query, the result includes only records that don't have **Null** values in the matching fields. For example, to list Suppliers and Customers who live in the same region, you create a query that includes the Suppliers and Customers tables, and the tables are joined on the Region field. When you view the results, you'll see only the values for records that contain a value in the Region field in both tables.

[Searching for Null values or zero-length strings](#)

If you're using a query to search for **Null** values or zero-length strings, type **Is Null** into the **Criteria** cell to search for **Null** values, or type two double quotation marks (" ") into the **Criteria** cell to search for zero-length strings (don't type a space between the quotation marks).

[Understanding how Null values affect numeric calculations](#)

If you use an [aggregate function](#) to calculate the sum, average, count, or another amount on a field's values, records with **Null** values in that field won't be included in the calculation. This is true whether you calculate the aggregate using the **Total** row in the query [design grid](#), the Simple Query Wizard, or a custom [expression](#). For example, if you use the **Count** function to count the number of values in a field, it will return a count of all the records with non-**Null**

values. If you want to find the total number of records including those with **Null** values, use **Count** with the asterisk (*) wildcard character.

Field:	TotalOrders: Count(*)
--------	-----------------------

If you use an arithmetic operator (such as +, -, *, /) in an expression (such as [UnitsInStock]+[UnitsOnOrder]), and one of the fields in the expression contains a **Null** value, then the result of the entire expression will be a **Null** value.

[Converting Null values to zero](#)

When you have fields that contain **Null** values, you can create an [expression](#) that converts the **Null** values to zero. You might do this if you want the records containing **Null** values to be included in an [aggregate](#) calculation, or if you want to prevent an expression from resulting in a **Null** value when a field that's referenced in the expression contains **Null** values. Use the **Nz** function, for example, to convert **Null** values to zero:

```
Nz([Subtotal],0+Nz([Freight],0))
```

[Combining Text fields that contain Null values](#)

If you're using an [expression](#) to combine two fields containing text values and one or both of the fields includes **Null** values, use the **&** operator instead of the + operator to combine the values. The **&** operator will combine the values even if they contain **Null** values, while the + operator will return a **Null** if either of the two values is a **Null**. For example:

```
FullName: [LastName]&" "&[FirstName]
```

[Sorting on fields that contain Null values and zero-length strings](#)

When you sort a field in ascending order, any records in which that field contains a **Null** value are listed first. If a field contains both **Null** values and zero-length strings, the **Null** values appear first in the sort order, immediately followed by the zero-length strings.

[Using a wildcard character to return non-Null values](#)

If you use the [expression](#) Like "*" (ANSI-89) or Like "%" (ANSI-92) when defining query [criteria](#) for a field, the query results will include zero-length strings in that field, but not **Null** values.

[Show All](#)

Select fields to add in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To select	Do this
A field	Click the field name.
A block of fields	Click the first field in the block, hold down SHIFT, and click the last field.
Noncontiguous fields	Hold down CTRL as you click the fields.
All fields	Double-click the title bar of the field list or click the asterisk (*).

[Show All](#)

Move a field in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#), or open a form or datasheet and show the [Advanced Filter/Sort window](#).
2. Do one of the following:

To select one column, click its [column selector](#).

To select more than one adjacent column, drag across their column selectors.

3. Click the selector of any one of the selected columns again, and then drag the field or fields to their new location.



- 1 A selected column
- 2 The pointer changes to a rectangle during the move

[Show All](#)

Show or hide a field in a query's results (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can add to the query [design grid](#) a field that you use for setting criteria or a sort order, yet not show the data from that particular field in the query's results.

1. Open the query in [Design view](#).
2. In the design grid, do one of the following:

To show the field, select the **Show** check box.

To hide the field, clear the **Show** check box.

Notes

- If you want to create a form, report, or [data access page](#) based on the query, and the query's **OutputAllFields** property is set to **No**, you must show the fields for which you will create [bound controls](#) on the form, report, or data access page.
- When you close a query, Microsoft Access moves hidden fields to the rightmost columns in the design grid.

[Show All](#)

Change a field name in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Changing a field name in the query [design grid](#) changes the column heading in query [Datasheet view](#). In addition, if you create a new object, such as a new form, report, or [data access page](#), based on the query, the new object uses the new field name. For example, if you create a [control](#) to display the renamed field, the new field name will be used for the control name and its label caption. The field name in the underlying table doesn't change, however, nor does it change in forms and reports that were based on the query before you changed the field name.

You can rename a field in a query to describe the field's data more accurately. This is particularly useful when you define a new [calculated field](#) or calculate a sum, count, or another type of total on an existing field. In such cases, unless you enter a name, Microsoft Access assigns names such as Expr1 or SumOfOrder Amount.

 [Rename a field](#)

Note Changing a field's name in the current query's design grid has no effect in query Datasheet view if the **Caption** property for that field has been set in the underlying table or query.

1. Open a query in [Design view](#).
2. Place the insertion point to the left of the first letter of the field name in the design grid.
3. Type the new name followed by a colon.



Note If you're replacing the field name Expr1 (or Expr2, and so on) that Access automatically assigned, replace only the name, not the expression that follows

the colon.

If you want the new field name to be used only for datasheet columns, or label captions in forms, reports, and data access pages, you can set the **Caption** property for the field instead of renaming the field in the design grid.

[Change a field's caption](#)

1. Open a query in [Design view](#).
2. In the design grid, click anywhere in the column of the field whose caption you want to change, and then click **Properties**  on the toolbar.
3. In the **Caption** property box, type the new caption for the field.

[Show All](#)

Customize the display of fields in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Automatically display all fields from a query's underlying table or query](#)

You can display all the fields without adding them to the [design grid](#).

1. Open a query in [Design view](#).
2. Select the query by clicking anywhere in query Design view outside the design grid and the [field lists](#).
3. Click **Properties**  on the toolbar to display the query's property sheet.
4. Set the **OutputAllFields** property to **Yes**.

Note If you want to limit the records that the query returns by specifying criteria, or if you want to create a [calculated field](#) or specify a sort order in the design grid, you need to add the fields for these purposes to the design grid. Make sure you clear the **Show** check box for any of these fields; otherwise, they will display twice in the query's results.

[Change the default setting for the **OutputAllFields** property](#)

This option determines whether the query's results automatically include all the fields from the underlying tables or queries (without adding them to the [design grid](#)), or only the fields you add to the design grid.

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab, and then select or clear the **Output All Fields** check box.

Note Changing this option only affects the property setting for new queries you create, not existing queries.

[Display a custom field description in a query's status bar](#)

1. Open a query in [Design view](#).
2. Click the field in the [design grid](#) whose description you want to change, and then click **Properties**  on the toolbar to display the field's property sheet.
3. In the **Description** property, type the description you want the status bar to display for that field in the query's results. The maximum length is 255 characters.

Users will see the new description in the status bar when they click in the field in query [Datasheet view](#).

[Show All](#)

Display a combo box from a table as a text box in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Changing how the field displays in the query doesn't affect how the field displays in the underlying table.

1. Open the query in [Design view](#).
2. In the query [design grid](#), right-click anywhere in the column of the field you want to change, and then click **Properties** on the shortcut menu.
3. Click the **Lookup** tab.
4. In the **DisplayControl** property box, click **Text Box**.

[Show All](#)

Enter criteria to affect when calculations are performed (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In the query [design grid](#), where you specify criteria determines when the calculation is performed and can produce different query results.

Note Although the following examples only show queries that calculate totals, these guidelines apply to [calculated fields](#) as well.

[Limit groups before performing calculations on groups of records](#)

- Specify the criteria in the **Group By** fields as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum 1
Sort:			
Show:			
Criteria:	"Canada"		
Or:	"UK"		

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Paillarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

- 1** This query totals extended prices for ...
- 2** ... companies in Canada and the UK only.

[Return selected results after calculations are performed](#)

- Specify criteria in the field that contains the calculation as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum
Sort:			
Show:			
Criteria:	"Canada"		<10000
or:	"UK"		<10000

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Pailarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

Ship Country	Company Name	SumOfExtendedPrice
Canada	Laughing Bacchus	\$522.50
UK	B's Beverages	\$7,383.90

- 1 This query totals extended prices for companies in Canada and the UK ...
- 2 ... but shows only those that are less than \$10,000.

[Limit records before they are grouped and before the calculation is performed](#)

- Add to the design grid the field whose records you want to limit, and then specify criteria in the field's **Criteria** cell.

If you're calculating totals in the same query, set the **Total** cell for the field containing the criteria to **Where**. This rule applies whether you're calculating the total on all records or groups of records. (Microsoft Access automatically clears the **Show** check box.)

The following example uses the ExtendedPrice field twice, once to limit the records and once to calculate the total. However, you can use a different field to limit records by dragging that field to the design grid and setting its **Total** cell to **Where**.

Ship Country	Company Name	ExtendedPrice
UK	Seven Seas Imports	\$345.00
Germany	Lehmanns Marktstand	\$525.00
UK	Seven Seas Imports	\$210.00
Canada	Mère Paillarde	\$691.20

1

Field:	ShipCountry	Company Name	ExtendedPrice	ExtendedPrice
Table:				
Total:	Group By	Group By	Sum	Where
Sort:				
Show:				
Criteria:	"Canada"			<500
or:	"UK"			

2

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$11,998.00
Canada	Mère Paillarde	\$21,691.36
UK	Around the Horn	\$4,907.50
UK	B's Beverages	\$1,380.00
UK	Eastern Connection	\$11,463.00
UK	Seven Seas Imports	\$3799.60

3

- 1 From these records, the query retrieves only those with extended prices greater than \$500.00 before it groups or totals ...
- 2 ... and then it sums and shows only those totals for companies in Canada or the UK.
- 3 This total for Seven Seas Imports does not include the order for \$210.00.

[Show All](#)

Add or remove a field in the design grid (MDB)

1. Open a query [Design view](#), or open a form, report, or datasheet and show the [Advanced Filter/Sort window](#).
2. Do one of the following:

[Add a field](#)

In a query, you add only those fields whose data you want to view, set criteria on, group by, update, or sort. In a filter, you add only the fields you want to use for sorting or specifying criteria, and Microsoft Access will automatically display all the fields in the filtered results.

1. Select one or more fields.

[How?](#)

To select	Do this
A field	Click the field name.
A block of fields	Click the first field in the block, hold down SHIFT, and click the last field.
Noncontiguous fields	Hold down CTRL as you click the fields.
All fields	Double-click the title bar of the field list or click the asterisk (*).

2. Drag the field from the [field list](#) to the column in the [design grid](#) where you want to insert it.

[Tip](#)

Instead of dragging, you can also add fields by double-clicking the name in the field list or selecting a field directly from the list box in the **Field** row on the grid.

[Remove a field](#)

- Select the field by clicking its [column selector](#), and then press the DELETE key.

Note When you remove a field from the [design grid](#), you're only removing it from the query or filter's design. You're not deleting the field and its data from the underlying table, nor, in the case of a filter, are you removing it from the filtered results.

[Show All](#)

Change a query's permissions (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Change default permissions for all new queries](#)

In a [security-enabled workgroup](#), you can change the default [permissions](#) that allow others to view data returned from queries, or, in the case of action queries, to run the queries, even if they are otherwise restricted from viewing the underlying table or query.

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab.
3. Click the **Run Permissions** option you want to use.

If you select **Owner's**:

- All users have the query [owner's](#) permission to view or run the query.
- Only the query owner can save changes to the query.
- Only the query owner can change the ownership of the query.

If you select **User's**, the permissions that are defined for that classification of users are in effect instead. Any user with Administer permissions for the query can save changes to the query or change its ownership.

Note Changing the default permissions affects only new queries.

[Permit others to view or run a query](#)

In a [security-enabled workgroup](#), you can assign others [permission](#) to view the data your query returns or, in the case of an action query, to run the query, even if they are otherwise restricted from viewing the query's underlying table or query.

By setting the **RunPermissions** property of the query to **Owner's**, all users have the query [owner's](#) permission to view or run the query, only the query owner can

save changes to the query, and only the query owner can change the ownership of the query.

Security Note Be careful using the **RunPermissions** property. In effect, you are giving users restricted read access to the data underlying the query. Sensitive or confidential information could be inadvertently revealed to other users.

1. Open the query in [Design view](#).
2. Select the query by clicking anywhere in query Design view outside the [design grid](#) and the [field lists](#).
3. Click **Properties**  on the toolbar to display the query's property sheet.
4. Set the **RunPermissions** property to **Owner's**.

[Show All](#)

Set multiuser query options (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Specify a record-locking strategy](#)

You can lock all records in the underlying table or only the record that you're editing. Either way, other users can't change the record in the underlying table while you are editing it in the query's results.

1. Open a query in [Design view](#). In a [multiuser](#) environment, confirm that all users have closed the query, the underlying table or query, and other objects based on the query.
2. Select the query by clicking anywhere in query Design view outside the [design grid](#) and the [field lists](#).
3. Click **Properties**  on the toolbar to display the query's property sheet.
4. Click the **RecordLocks** property setting you want to use.

[Specify how long before a query displays an ODBC time-out error](#)

A query that uses an [ODBC](#) database as its underlying source of data might encounter delays due to network traffic or heavy use of the ODBC server.

1. Open a query in [Design view](#).
2. Select the query by clicking anywhere in query Design view outside the [design grid](#) and the [field lists](#).
3. Click **Properties**  on the toolbar to display the query's property sheet.
4. Set the **ODBCTimeout** property to the number of seconds that the query waits before displaying a time-out error. The default is 60 seconds.

Note If you set the property to zero, no time-out error occurs.

[Show All](#)

About joining tables or queries in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The power of queries lies in being able to bring together or perform an action on data from more than one table or query. For example, you might want to view a customer's information with the orders the customer placed. To see this information, you need data from the Customers and Orders tables.

Customers : Table	
Company Name	Customer ID
Ernst Handel	ERNSH
Familia Arquibaldo	FAMIA
Folies gourmandes	FOLIG

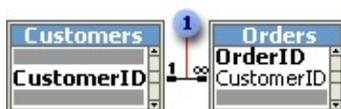
Orders : Table	
Customer ID	Order ID
ERNSH	11072
FISSA	10184
FOLIG	10016

Customers Orders : Select Query		
Company Name	Customer ID	Order ID
Ernst Handel	ERNSH	11072
Folies gourmandes	FOLIG	10016

- 1 You can combine records from two tables ...
- 2 ... by creating a query that shows just the information you need.

[Joins and relationships](#)

When you add more than one table or query to a query, you need to make sure their [field lists](#) are joined to each other with a [join](#) line so that Microsoft Access knows how to connect the information.



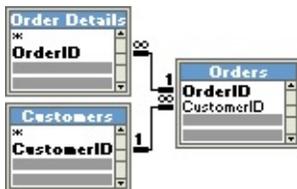
- 1 Join line

If you previously created [relationships](#) between tables in the [Relationships window](#), Access automatically displays join lines when you add related tables in query [Design view](#). If [referential integrity](#) is enforced, Access also displays a "1" above the join line to show which table is on the "one" side of a [one-to-many relationship](#) and an infinity symbol ∞ to show which table is on the "many" side.

[Joins not defined in relationships](#)

Even if you haven't created [relationships](#), Microsoft Access automatically creates [joins](#) if you add two tables to a query and the tables each have a field with the same or compatible [data type](#) and if one of the join fields is a [primary key](#). The "one" and "many" symbols are not displayed in this case, because [referential integrity](#) is not enforced.

Sometimes the tables you add to the query don't include any fields that can be joined. In this situation, you have to add one or more extra tables or queries to serve solely as a bridge between the tables whose data you want to use. For example, if you add the Customers and Order Details tables to a query, they won't have a join line between them because they don't have any fields that can be joined. But the Orders table is related to both tables, so you can include the Orders table in your query to provide a connection between the other two.



[Types of joins](#)

[Inner joins](#)

Once tables and queries are joined, and you've added fields from both tables or queries to the [design grid](#) in query [Design view](#), the default [join](#) tells the query to check for matching values in the join fields. (This is called an [inner join](#).) When it finds matches, it combines those two records and displays them as one record in the query's results.



1 The symbols above the join line indicate the type of relationship, in this case, "one" to "many."

2 If one table doesn't have a matching record in the other table, neither record appears in the query's results.

3 Inner join

Outer joins

If one table or query doesn't have a matching record in the other table or query, neither record appears in the query's results. If you want the query to select all the records from one table or query whether or not it has matching records in the other table or query, you can change the [join](#) type to an [outer join](#).



1 An arrow points to the table contributing only the matching records.

2 The query displays empty cells where there is no matching record from

another table.

3 Outer join

[Unequal joins](#)

If you want the query to select records based on the value in the [join](#) field being greater than, less than, not equal to, greater than or equal to, or less than or equal to the value in the other join field, you must create an [SQL statement](#) in [SQL view](#).

[Cross-product or Cartesian product joins](#)

If tables in a query aren't joined to one another, either directly or indirectly, Microsoft Access doesn't know which records are associated with which, so it displays every combination of records between the two tables. Therefore, if each table had 10 records in it, the query's results will contain 100 records (10X10). This result set of every possible combination is called a cross product or Cartesian product. These queries might take a long time to run and ultimately might produce less meaningful results.

Create a lookup column

[Create an AutoLookup query \(MDB\)](#)

[Create a field that looks up or lists values in tables \(MDB\)](#)

[Create a lookup column \(ADP\)](#)

[Show All](#)

"many" side of the one-to-many relationship. (In other words, it's the [foreign key](#) for that table.) For AutoLookup to work, this field can't be a primary key and its **Indexed** property can't be set to **Yes (No Duplicates)**. For example, in a query that includes the Customers and Orders tables, drag to the design grid the CustomerID field from the Orders table, not the one from the Customers table.

- The value you enter in the join field from the "many" side must already exist in the join field from the "one" side.

[What happens when you update data](#)

When the value of the join field from the "many" side of the relationship is added or changed in a record, Access automatically finds and displays the associated values from the table on the "one" side of the relationship.

You can always update the join field from the "many" side of a relationship, but you can update the join field from the "one" side only if you enabled [cascading updates](#) when defining the relationship between the tables. Either way, when you update data, Access automatically recalculates any totals or expressions in the query that are dependent on the updated data.

[Show All](#)

Enable or disable automatic joins in queries (MDB)

Even if you haven't created [relationships](#) between tables, Microsoft Access automatically creates joins if you add two tables to a query and the tables each have a field with the same or compatible [data type](#) and if one of the join fields is a [primary key](#). You can enable or disable these automatic joins.

1. On the **Tools** menu, click **Options**.
2. Click the **Tables/Queries** tab.
3. Select or clear the **Enable AutoJoin** check box.

Note This setting applies to new queries only.

[Show All](#)

Join tables and queries in a query (MDB)

In most cases, you want to bring together or perform an action on data from more than one table or query. For example, you might want to view a customer's information with the orders the customer placed. To see this information, you need data from the Customers and Orders tables.

[Join different tables and queries](#)

- In query [Design view](#), drag a field from the [field list](#) for one table or query to the equivalent field (a field of the same or compatible [data type](#) containing similar data) in the field list for the other table or query.



With this type of [join](#), Microsoft Access selects records from both tables or queries only when the values in the joined fields are equal.

Note Join numeric fields only if the **FieldSize** property settings for both are **Byte**, **Integer**, or **Long Integer**.

In some cases, you want to join two copies of the same table or query, called a [self-join](#), that combines records from the same table when there are matching values in the joined fields. For example, say you have an Employees table in which the ReportsTo field for each employee's record displays his or her manager's ID instead of name. You could use a self-join to display the manager's name in each employee's record instead.

[Join two copies of the same table or query](#)

1. In query [Design view](#), add the table to the query twice.
2. Create the [join](#) by dragging a field in the first table's [field list](#) to the field you want to relate it to in the second field list.

Note Microsoft Access appends "_1" to the table name in the second field list. For example, if you add the Employees table twice, the first field list is titled "Employees," and the second field list is titled "Employees_1." You can rename the table to something more descriptive by setting the **Alias** or **Caption** property for the field list.

[Example of a self-join using the Employees table](#)

To display the managers' names in the ReportsTo field instead of the managers' IDs, follow these steps.

1. In query Design view, add the Employees table to the query twice.
2. Rename the second table in the query.

[How?](#)

Note Renaming copies of a table or query in a query doesn't rename the underlying table or query.

1. Open a query in [Design view](#).
2. Click anywhere in the second copy of the [field list](#) for the table or query, and then click **Properties**  on the toolbar to display the property sheet.
3. In the **Alias** property box, type a new name for the table or query.
3. Create the join by dragging the ReportsTo field in the first table's field list to the EmployeeID field in the second table's field list.
4. Add the LastName, FirstName, and Title fields from the first field list to the [design grid](#).
5. Add the LastName field from the second field list to the design grid. To display the field name as "Manager" instead of "LastName," set the **Caption** property in this field's property sheet.

[Show All](#)

Create an AutoLookup query (MDB)

[For AutoLookup to work on new records, certain conditions must be met.](#)

For AutoLookup to work, certain conditions must be met:

- The query must be based on more than one table and the tables must have a [one-to-many relationship](#). ([Referential integrity](#) doesn't have to be enforced.)
 - The [join](#) field on the "one" side of the relationship must have a [unique index](#). A unique index means that the field is a [primary key](#) or its **Indexed** property in table [Design view](#) is set to **Yes (No Duplicates)**.
 - The join field you add to the [design grid](#) must come from the table on the "many" side of the one-to-many relationship. (In other words, it's the [foreign key](#) for that table.) For AutoLookup to work, this field can't be a primary key and its **Indexed** property can't be set to **Yes (No Duplicates)**. For example, in a query that includes the Customers and Orders tables, drag to the design grid the CustomerID field from the Orders table, not the one from the Customers table.
 - The value you enter in the join field from the "many" side must already exist in the join field from the "one" side.
1. Create a [select query](#), adding two tables that have a [one-to-many relationship](#).

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.

6. To view the query's results, click **View**  on the toolbar.
2. Add the fields for the query to the [design grid](#).

You must include the join field from the "many" side of the one-to-many relationship.



- 1 Use the join field ...
- 2 ... from the "many" table.
3. To retrieve specific records, specify criteria in the **Criteria** row.

[How?](#)

1. Open a query in [Design view](#), or display the [Advanced Filter/Sort window](#) for a table, query, or form.
2. Click the first **Criteria** cell for the field you want to set criteria for.
3. Enter the criteria [expression](#) by typing it or by using the Expression Builder. To display the Expression Builder, right-click the **Criteria** cell, and then click **Build**.

Field:	ShippedDate	ShipName
Sort:		
Show:		
Criteria:	#8/11/2001#	"BonApp"

- 1 Retrieves orders shipped to Bon App' on 8/11/2001.

If your query includes linked tables, the values you specify in criteria on fields from the linked tables are [case-sensitive](#)— they must match the case of the values in the underlying table.

4. To enter another expression in the same field or in another field, move to the appropriate **Criteria** cell and enter the expression.

4. To view the query's results, click **View**  on the toolbar.

When you enter new data in the join field on the "many" side of the relationship, Microsoft Access will automatically look up and fill in the corresponding data from the "one" side.

[Show All](#)

Delete a join line in a query (MDB)

- In query [Design view](#), select the join line, and then press the DELETE key.

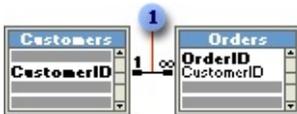


- 1 Join line

[Show All](#)

Change the type of join in a query (MDB)

1. Open a query in [Design view](#).
2. Double-click the join line between the [field lists](#) for the tables or queries.



1 Join line

3. In the **Join Properties** dialog box, click the join option you want, and then click **OK**.

Note From the [Relationships window](#), you can change the default join type for tables.

[Show All](#)

Enter criteria to retrieve certain records (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#), or display the [Advanced Filter/Sort window](#) for a table, query, or form.
2. Click the first **Criteria** cell for the field you want to set criteria for.
3. Enter the criteria [expression](#) by typing it or by using the Expression Builder. To display the Expression Builder, right-click the **Criteria** cell, and then click **Build**.

Field:	ShippedDate	ShipName
Sort:		
Show:		
Criteria:	#8/11/2001#	"BonApp"

1

- 1 Retrieves orders shipped to Bon App' on 8/11/2001.

If your query includes linked tables, the values you specify in criteria on fields from the linked tables are [case-sensitive](#)— they must match the case of the values in the underlying table.

4. To enter another expression in the same field or in another field, move to the appropriate **Criteria** cell and enter the expression.

[Show All](#)

Examples of expressions

You use [expressions](#) in many operations in Microsoft Access, including creating [calculated controls](#), query and filter criteria, default values, validation rules, and macro conditions. Often, the easiest way to create an expression is to find an example of an expression that is similar to what you want to create, and then modify it to do what you want.

 [Examples of expressions used in forms, reports, and data access pages](#)

 [Examples of manipulating and calculating dates](#)

The following table lists examples of expressions that you can use in [calculated controls](#) on forms, reports, and [data access pages](#).

Expression	Description
=Date()	Uses the Date function to display the current date in the form of <i>mm-dd-yyyy</i> , where <i>mm</i> is the month (1 through 12), <i>dd</i> is the day (1 through 31), and <i>yyyy</i> is the year (1980 through 2099).
=Format(Now(), "ww")	Uses the Format function to display the number of the week of the year the current date represents, where <i>ww</i> is 1 through 53.
=DatePart("yyyy", [OrderDate])	Uses the DatePart function to display the four-digit year of the value of the OrderDate field.
=DateAdd("y", -10, [PromisedDate])	Uses the DateAdd function to display a date that is 10 days before the value of the PromisedDate field.
=DateDiff("d", [OrderDate], [ShippedDate])	Uses the DateDiff function to display the variance in days between the values of the OrderDate and ShippedDate fields.

Notes

- In the **ControlSource** property of a calculated control, precede the

expression with the = operator. On a data access page, you can omit the = operator, and type an alias instead; for example, type **WeekNumber: Format(Now(), "ww")**.

- When you set the **Name** property of a calculated control in a form or report, or set the **ID** property of a calculated control in a data access page, make sure that you use a unique name. Don't use the name or ID of one of the controls you used in the expression.
- In an expression on a form or report, you can use the name of a control or the name of a field in the underlying records source. In an expression on a data access page, you can use only the name of a field that's in the [data definition](#) of the page.

[Examples of manipulating text values](#)

The following table lists examples of expressions that you can use in [calculated controls](#) on forms, reports, and [data access pages](#).

Expression	Description
= "N/A"	Displays N/A.
= [FirstName] & " " & [LastName]	Displays the values of the FirstName and LastName fields separated by a space.
= Left([ProductName], 1)	Uses the Left function to display the first character of the value of the ProductName field.
= Right([AssetCode], 2)	Uses the Right function to display the last 2 characters of the value of the AssetCode field.
= Trim([Address])	Uses the Trim function to display the value of the Address field, removing any leading or trailing spaces.
= IIf(IsNull([Region]), [City] & " " & [PostalCode], [City] & " " & [Region] & " " & [PostalCode])	Uses the IIf function to display the values of the City and PostalCode fields if Region is Null ; otherwise, it displays the values of the City, Region, and PostalCode fields, separated by spaces.

Notes

- In the **ControlSource** property of a calculated control, precede the expression with the = operator. On a data access page, you can omit the =

operator, and type an alias instead; for example, type **FullName: [FirstName]&" "& [LastName]**.

- When you set the **Name** property of a calculated control in a form or report, or set the **ID** property of a calculated control in a data access page, make sure to use a unique name. Don't use the name or ID of one of the controls that you used in the expression.
- In an expression on a form or report, you can use the name of a control or field in the underlying record source. In an expression on a data access page, you can use only the name of a field that's in the [data definition](#) of that page.

[Examples of manipulating arithmetic operations](#)

The following table lists examples of expressions that you can use in [calculated controls](#) on forms, reports, and [data access pages](#).

Expression	Description
= $[\text{Subtotal}] + [\text{Freight}]$	The sum of the values of the Subtotal and Freight fields.
= $[\text{RequiredDate}] - [\text{ShippedDate}]$	The difference between the values of the RequiredDate and ShippedDate fields.
= $[\text{Price}] * 1.06$	The product of the value of the Price field and 1.06 (adds 6 percent to the Price value).
= $[\text{Quantity}] * [\text{Price}]$	The product of the values of the Quantity and Price fields.
= $[\text{EmployeeTotal}] / [\text{CountryTotal}]$	The quotient of the values of the EmployeeTotal and CountryTotal fields.

Notes

- In the **ControlSource** property of a calculated control, precede the expression with the = operator. On a data access page, you can omit the = operator, and type an alias instead; for example, type **ExtendedPrice: [Quantity] * [Price]**.
- When you set the **Name** property of a calculated control in a form or report, or set the **ID** property of a calculated control on a data access page, make sure to use a unique name. Don't use the name or ID of one of the controls that you used in the expression.
- In an expression on a form or report, you can use the name of a control or

the name of a field in the underlying record source. In an expression on a data access page, you can use only the name of a field that's in the [data definition](#) of the page.

- When you use an arithmetic operator (+, -, *, /) in an expression, and the value of one of the controls in the expression is **Null**, the result of the entire expression will be **Null**. On a form or report, if some records in one of the controls that you used in the expression might have a **Null** value, you can convert the **Null** value to zero by using the **Nz** function; for example:

```
=Msodsc.Nz([Subtotal]) + Msodsc.Nz([Freight])
```

[Examples of conditional expressions](#)

The following table lists examples of expressions that you can use in [calculated controls](#) on forms, reports, and [data access pages](#).

Expression	Description
=IIf([Confirmed] = "Yes", "Order Confirmed", "Order Not Confirmed")	Displays the message "Order Confirmed" if the value of the Confirmed field is Yes; otherwise, it displays the message "Order Not Confirmed."
=IIf(IsNull([Country]), " ", [Country])	Displays an empty string if the value of the Country field is Null ; otherwise, it displays the value of the Country field.
=IIf(IsNull([Region]), [City]&" "& [PostalCode], [City]&" "&[Region]&" "&[PostalCode])	Display the values of the City and PostalCode fields if Region is Null ; otherwise, it displays the values of the City, Region, and PostalCode fields.
=IIf(IsNull([RequiredDate] - [ShippedDate]), "Check for a missing date", [RequiredDate] - [ShippedDate])	Display the message "Check for a missing date" if the result of subtracting ShippedDate from RequiredDate is Null ; otherwise, it displays the difference between the values of the RequiredDate and ShippedDate fields.

Notes

- In the **ControlSource** property of a calculated control, precede the expression with the = operator. On a data access page, you can omit the = operator, and type an alias instead; for example, type **DisplayCountry**:

IIf(IsNull([Country])," ",[Country]).

- When you set the **Name** property of a calculated control in a form or report, or set the **ID** property of a calculated control in a data access page, make sure to use a unique name. Don't use the name or ID of one of the controls that you used in the expression.
- In an expression on a form or report, you can use the name of a control or the name of a field in the underlying record source. In an expression on a data access page, you can use only the name of a field that's in the [data definition](#) of the page.

[Examples of aggregate and domain aggregate functions](#)

Note You can't use aggregate and domain aggregate functions in a [data access page](#). Additionally, you can't use domain aggregate functions in a [Microsoft Access project \(.adp\)](#).

[Examples of expressions that use aggregate functions](#)

Expression	Description
=Avg([Freight])	Uses the Avg function to display the average of the values of the Freight control.
=Count([OrderID])	Uses the Count function to display the number of records in the OrderID control.
=Sum([Sales])	Uses the Sum function to display the sum of the values of the Sales control.
=Sum([Quantity] * [Price])	Uses the Sum function to display the sum of the product of the values of the Quantity and Price controls.
=([Sales] / Sum([Sales]) * 100	Displays the percentage of sales, determined by dividing the value of the Sales control by the sum of all the values of the Sales control. Note If the control's Format property is set to Percent , don't include the *100.

[Examples of expressions that use domain aggregate functions](#)

Expression	Description
=DLookup("[ContactName]", "[Suppliers]", "[SupplierID] = Forms![SupplierID]")	Uses the DLookup function to display the value of the ContactName field in the Suppliers table where the value of the SupplierID field in the table matches the value of the SupplierID control on the active form.
=DLookup("[ContactName]", "[Suppliers]", "[SupplierID] = Forms![New Suppliers]![SupplierID]")	Uses the DLookup function to display the value of the ContactName field in the Suppliers table where the value of the SupplierID field in the table matches the value of the SupplierID control on the New Suppliers form.
=DSum("[OrderAmount]", "[Orders]", "[CustomerID] = 'RATTC'")	Uses the DSum function to display the sum total of values of the OrderAmount field in the Orders table where the CustomerID is RATTC.

Notes

- In a calculated control, precede the expression with the = operator.
- When you set the **Name** property of a calculated control, make sure you use a unique name. Don't use the name of one of the controls you used in the expression.

[Examples of expressions used in queries and filters](#)

[Examples of criteria used to retrieve records](#)

[Ranges of values \(>, <, >=, <=, <>, or Between...And\)](#)

Expression	Result
> 234	For a Quantity field, numbers greater than 234
< 1200.45	For a UnitPrice field, numbers less than 1200.45
>= "Callahan"	For a LastName field, all names from Callahan through the end of the alphabet
Between #2/2/1999# And	For an OrderDate field, dates from 2-Feb-99 through

#12/1/1999#	1-Dec-99 (ANSI-89)
Between '2/2/1999' And '12/1/1999'	For an OrderDate field, dates from 2-Feb-99 through 1-Dec-99 (ANSI-92)

[Values that don't match \(Not\)](#)

Expression	Result
Not "USA"	For a ShipCountry field, orders shipped to countries other than the USA
Not 2	For an ID field, the employee whose ID doesn't equal 2
Not T*	For a LastName field, employees whose names don't start with the letter "T" (ANSI-89)
Not T%	For a LastName field, employees whose names don't start with the letter "T" (ANSI-92)

[Values in a list \(In\)](#)

Expression	Result
In("Canada", "UK")	For a ShipCountry field, orders shipped to Canada or the UK
In(France, Germany, Japan)	For a CountryName field, employees living in France or Germany or Japan

[Text, partial, and matching values](#)

Expression	Result
"London"	For a ShipCity field, orders shipped to London
"London" Or "Hedge End"	For a ShipCity field, orders shipped to London or Hedge End
>="N"	For a CompanyName field, orders shipped to companies whose name starts with the letters N through Z
Like "S*"	For a ShipName field, orders shipped to customers whose name starts with the letter S (ANSI-89)
Like "S%"	For a ShipName field, orders shipped to customers whose name starts with the letter S (ANSI-92)

Right([OrderID], 2) = For an OrderID field, orders with ID values ending in 99
"99"

Len([CompanyName]) > Val(30) For a CompanyName field, orders for companies whose
name is more than 30 characters long

[Part of a field's value \(Like\)](#)

Expression	Result
Like "S*"	For a ShipName field, orders shipped to customers whose names start with the letter S (ANSI-89)
Like "S%"	For a ShipName field, orders shipped to customers whose names start with the letter S (ANSI-92)
Like "*Imports"	For a ShipName field, orders shipped to customers whose names end with the word "Imports" (ANSI-89)
Like "%Imports"	For a ShipName field, orders shipped to customers whose names end with the word "Imports" (ANSI-92)
Like "[A-D]*"	For a ShipName field, orders shipped to customers whose names start with A through D (ANSI-89)
Like "[A-D]%"	For a ShipName field, orders shipped to customers whose names start with A through D (ANSI-92)
Like "*ar*"	For a ShipName field, orders shipped to customers whose names include the letter sequence "ar" (ANSI-89)
Like "%ar%"	For a ShipName field, orders shipped to customers whose names include the letter sequence "ar" (ANSI-92)
Like "Maison Dewe?"	For a ShipName field, orders shipped to the customer with "Maison" as the first part of its name and a 5-letter second name in which the first 4 letters are "Dewe" and the last letter is unknown (ANSI-89)
Like "Maison Dewe_"	For a ShipName field, orders shipped to the customer with "Maison" as the first part of its name and a 5-letter second name in which the first 4 letters are "Dewe" and the last letter is unknown (ANSI-92)

[Dates](#)

Expression

Result

#2/2/2000#	For a ShippedDate field, orders shipped on February 2, 2000 (ANSI-89)
'2/2/2000'	For a ShippedDate field, orders shipped on February 2, 2000 (ANSI-92)
Date()	For a RequiredDate field, orders for today's date
Between Date() And DateAdd("M", 3, Date())	For a RequiredDate field, orders required between today's date and three months from today's date
< Date() - 30	For an OrderDate field, orders more than 30 days old
Year([OrderDate]) = 1999	For an OrderDate field, orders with order dates in 1999
DatePart("q", [OrderDate]) = 4	For an OrderDate field, orders for the fourth calendar quarter
DateSerial(Year ([OrderDate]), Month([OrderDate]) + 1, 1) - 1	For an OrderDate field, orders for the last day of each month
Year([OrderDate]) = Year(Now()) And Month([OrderDate]) = Month(Now())	For an OrderDate field, orders for the current year and month

[A blank field's value \(Null or zero-length string\)](#)

Expression	Result
Is Null	For a ShipRegion field, orders for customers whose ShipRegion field is Null (blank)
Is Not Null	For a ShipRegion field, orders for customers whose ShipRegion field contains a value
" "	For a Fax field, orders for customers who don't have a fax machine, indicated by a zero-length string value in the Fax field instead of a Null (blank) value

[The result of a domain aggregate function](#)

Expression	Result
> (DStDev("[Freight]",	For a Freight field, orders for which the freight

"Orders") + DAvg("[Freight]", "Orders")) cost rose above the mean plus the standard deviation for freight cost
 >DAvg("[Quantity]", "Order Details") For a Quantity field, products ordered in quantities above the average order quantity

[The result of a subquery as criteria](#)

Expression	Result
(SELECT [UnitPrice] FROM [Products] WHERE [ProductName] = "Aniseed Syrup")	For a UnitPrice field, products whose price is the same as the price of Aniseed Syrup
> (SELECT AVG([UnitPrice]) FROM [Products])	For a UnitPrice field, products that have a unit price above the average
> ALL (SELECT [Salary] FROM [Employees] WHERE ([Title] LIKE "*Manager*") OR ([Title] LIKE "*Vice President*"))	For a Salary field, the salary of every sales representative whose salary is higher than that of all employees with "Manager" or "Vice President" in their titles
> ALL (SELECT AVG([UnitPrice] * [Quantity]) FROM [Order Details])	For an OrderTotal: [Unit Price] * [Quantity] <u>calculated field</u> , orders with totals that are higher than the average order value

[Examples of expressions used in calculated fields](#)

[Manipulate text values](#)

Expression	Result
FullName: [FirstName] & " " & [LastName]	Displays in the FullName field the value of the FirstName and LastName fields, separated by a space
Address2: [City] & " " & [Region] & " " & [PostalCode]	Displays in the Address2 field the value of the City, Region, and PostalCode fields, separated by spaces
ProductInitial: Left([ProductName], 1)	Displays in the ProductInitial field the first character of the value in the ProductName field
TypeCode: Right([AssetCode], 2)	Displays in the TypeCode field the last two characters of the value in the AssetCode field

AreaCode: Mid([Phone], 2, 3) Displays in the AreaCode field the three characters starting with the second character of the value in the Phone field

[Perform arithmetic operations](#)

Expression	Result
PrimeFreight: [Freight] * 1.1	Displays in the PrimeFreight field freight charges plus 10 percent
OrderAmount: [Quantity] * [UnitPrice]	Displays in the OrderAmount field the product of the values in the Quantity and UnitPrice fields
LeadTime: [RequiredDate] - [ShippedDate]	Displays in the LeadTime field the difference between the values in the RequiredDate and ShippedDate fields
TotalStock: [UnitsInStock] + [UnitsOnOrder]	Displays in the TotalStock field the sum of the values in the UnitsInStock and UnitsOnOrder fields
FreightPercentage: Sum([Freight]) / Sum([Subtotal]) * 100	<p>Displays in the FreightPercentage field the percentage of freight charges in each subtotal, by dividing the sum of the values in the Freight field by the sum of the values in the Subtotal field.</p> <p>The Total row in the design grid must be displayed, and the Total cell for this field must be set to Expression.</p> <p>If the Format property of the field is set to Percent, don't include the *100.</p>
SumofUnits: Nz([UnitsInStock], 0) + Nz([UnitsOnOrder], 0)	<p>Displays in the SumofUnits field the sum of all units in stock and on order.</p> <p>When you use an arithmetic operator (+, -, *, /) in an expression and the value of one of the fields in the expression is Null, the result of the entire expression will be Null. If some records in one of the fields you used in the expression might have a Null value, you can convert the Null value to zero using the Nz function.</p>

[Manipulate and calculate dates](#)

Expression	Result
LagTime: DateDiff("d", [OrderDate], [ShippedDate])	Display in the LagTime field the number of days between the order date and ship date
YearHired: DatePart("yyyy", [HireDate])	Displays in the YearHired field the year each employee was hired
MonthNo: DatePart("M", [OrderDate])	Displays in the MonthNo field the number of the month
PriorDate: Date() - 30	Displays in the PriorDate field the date 30 days prior to the current date

[Use SQL and domain aggregate functions](#)

Expression	Result
Count(*)	Uses the Count function to count the number of records in the query, including records with Null (blank) fields
FreightPercentage: Sum([Freight]) / Sum([Subtotal]) * 100	Displays in the FreightPercentage field the percentage of freight charges in each subtotal, by dividing the sum of the values in the Freight field by the sum of the values in the Subtotal field. The Total row in the design grid must be displayed, and the Total cell for this field must be set to Expression . If the Format property of the field is set to Percent , don't include the *100.
AverageFreight: DAvg("[Freight]", "[Orders]")	Displays in the AverageFreight field the average discount given on all orders combined in a totals query

[Work with Null values](#)

Expression	Result
------------	--------

CurrentCountry: IIf(IsNull([Country]), " ", [Country])	Displays in the CurrentCountry field an empty string if the Country field is Null ; otherwise, it displays the contents of the Country field
LeadTime: IIf(IsNull([RequiredDate] - [ShippedDate]), "Check for a missing date", [RequiredDate] - [ShippedDate])	Displays in the LeadTime field the message "Check for a missing date" if the value of either the RequiredDate or ShippedDate fields is Null ; otherwise, it displays the difference
SixMonthSales: Nz([Qtr1Sales], 0) + Nz([Qtr2Sales], 0)	Displays in the SixMonthSales field the total of the values in the first-quarter and second-quarter sales fields combined, using the Nz function to convert the Null values to zero first

[Use a subquery](#)

Expression	Result
Category: (SELECT [CategoryName] FROM [Categories] WHERE [Products]. [CategoryID] = [Categories]. [CategoryID])	Displays in the Category field the CategoryName if the CategoryID from the Categories table is the same as the CategoryID from the Products table

[Examples of expressions used in update queries](#)

Use [expressions](#) such as the following in the **Update To** cell in the query [design grid](#) for the field you want to update.

Expression	Result
"Salesperson"	Changes a text value to Salesperson
#8/10/99#	Changes a date value to 10-Aug-99
Yes	Changes a No value in a Yes/No field to Yes
"PN" & [PartNumber]	Adds PN to the beginning of each specified part number
[UnitPrice] * [Quantity]	Calculates the product of UnitPrice and Quantity
[Freight] * 1.5	Increases freight charges by 50 percent
DSum("[Quantity] *	

[UnitPrice]", "Order Details", " [ProductID]=" & [ProductID])	Where the Product IDs in the current table match the Product IDs in the Order Details table, updates sales totals based on the product of Quantity and UnitPrice
Right([ShipPostalCode], 5)	Truncates the leftmost characters, leaving the five rightmost characters
IIf(IsNull([UnitPrice]), 0, [UnitPrice])	Changes a Null value to a zero (0) in the UnitPrice field

 [Examples of expressions used in SQL statements](#)

You can use an [expression](#) in many places in an [SQL statement](#), as the following examples show. Expressions are shown in bold text.

Expression	Result
SELECT [FirstName], [LastName] FROM [Employees] WHERE [LastName] = "Davolio";	Displays the values in the FirstName and LastName fields for employees whose last name is Davolio.
SELECT [ProductID], [ProductName] FROM [Products] WHERE [CategoryID] = Forms! [New Products]![CategoryID];	Displays the values in the ProductID and ProductName fields in the Products table for records in which the CategoryID value matches the CategoryID value specified in an open New Products form.
SELECT Avg([ExtendedPrice]) AS [Average Extended Price] FROM [Order Details Extended] WHERE [ExtendedPrice] > 1000;	Displays in a field named Average Extended Price the average extended price of orders for which the value in the ExtendedPrice field is more than 1,000.
SELECT [CategoryID], Count([ProductID]) AS [CountOfProductID] FROM [Products] GROUP BY [CategoryID] HAVING Count([ProductID]) > 10;	Displays in a field named CountOfProductID the total number of products for categories with more than 10 products.

 [Examples of default value expressions](#)

Expression	Default field value
-------------------	----------------------------

1	1
"MT"	MT
"New York, N.Y."	New York, N.Y. (note that you must enclose the value in quotes if it includes punctuation)
" "	Zero-length string
Date()	Today's date
=Yes	Yes displayed in the local language of the computer.

[Examples of macro condition expressions](#)

You can use any [expression](#) that evaluates to True/False or Yes/No in a macro [condition](#). The macro will be executed if the condition evaluates to True (or Yes).

[Tip](#)

To cause Microsoft Access to temporarily ignore an [action](#), enter False as a condition. Temporarily ignoring an action can be helpful when you are trying to find problems in a macro.

Use this expression	To carry out the action if
[City]="Paris"	Paris is the City value in the field on the form from which the macro was run.
DCount("[OrderID]", "Orders")>35	There are more than 35 entries in the OrderID field of the Orders table.
DCount("*", "Order Details", "[OrderID]=Forms![Orders]![OrderID]")>3	There are more than three entries in the Order Details table for which the OrderID field of the table matches the OrderID field on the Orders form.
[ShippedDate] Between #2-Feb-2001# And #2-Mar-2001#	The value of the ShippedDate field on the form from which the macro is run is no earlier than 2-Feb-2001 and no later than 2-Mar-2001.
Forms![Products]![UnitsInStock]<5	The value of the UnitsInStock field on the Products form is less than five.
IsNull([FirstName])	The FirstName value on the form from which the macro is run is Null (has no value). This expression is

	equivalent to [FirstName] Is Null .
[Country]="UK" And Forms![SalesTotals]! [TotalOrds]>100	The value in the Country field on the form from which the macro is run is UK, and the value of the TotalOrds field on the SalesTotals form is greater than 100.
[Country] In ("France", "Italy", "Spain") And Len([PostalCode])<>5	The value in the Country field on the form from which the macro is run is France, Italy, or Spain, and the postal code isn't five characters long.
MsgBox("Confirm changes?",1)=1	You click OK in a dialog box in which the MsgBox function displays "Confirm changes?". If you click Cancel in the dialog box, Microsoft Access ignores the action.

[Show All](#)

Use the asterisk (*) in the query design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#).
2. Drag the asterisk (*) from the field list to the **Field** row in the [design grid](#).
3. Add the fields you want to sort or set criteria for by dragging them from the field list to the **Field** row.
4. Select a sort order in the **Sort** cell for the fields you want to sort.
5. Enter criteria in the **Criteria** row for the appropriate fields.
6. Clear the **Show** check box for all fields except the asterisk; otherwise those fields will appear twice in the query's results.
7. Click **View**  on the toolbar to view the results.

[Show All](#)

Insert or delete a criteria row in the design grid (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Open a query in [Design view](#), or open a form or datasheet and show the [Advanced Filter/Sort window](#).
2. Do one of the following:

To insert a criteria row, click in the row that is below where you want the new row to appear, and then click **Rows** on the **Insert** menu. A new row is inserted above the row that you clicked.

To delete a criteria row, click anywhere in the row, and then click **Delete** on the **Edit** menu.

[Show All](#)

Use expressions in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

To use an expression in a query, you can type it directly into the **Grid** pane or you can enter it in the SQL pane as part of the statement. Entering expressions is similar to entering column names.

[Tip](#)

To make it easier to see long expressions on your screen, you can resize the columns in the **Grid** pane.

[Displaying expressions in the result set](#)

You can display an expression in the result set by specifying the expression in place of a column in the **Grid** pane.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Grid** pane, insert a new grid row.
3. In the column of the new grid row, type the expression whose results you want to display.

Column	Alias	Table	Output
lname + ', ' + fname	full_name		<input checked="" type="checkbox"/>
			<input type="checkbox"/>

When you display the results of an expression in the result set, the database assigns a column heading to it using the format "Expr*n*," where *n* is the number of the expression in the current query. You can replace this with a more meaningful alias for the expression.

[Sorting with expressions](#)

You can sort by the results of an expression. As with columns, you specify the

sort type and sort order.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Grid** pane, insert a new grid row.
3. In the column of the new grid row, type the expression you want to sort by.
4. If you do not want to display the expression in the query, clear the **Output** column of the new row.
5. In the **Sort Type** column, choose **Ascending** or **Descending**, and then in the **Sort Order** column, choose the sort priority for the expression.

Column	Alias	Table	Output	Sort Type	Sort Order
sales.qty * titles.price			<input type="checkbox"/>	Ascending	1

[Searching with expressions](#)

There are two ways to use an expression for searching. The expression can be the condition against which you compare values, or it can function as the value you are comparing.

[Expression examples](#)

The following example illustrates how you can use an expression as the condition in a WHERE clause:

```
SELECT ord_num, ord_date
FROM sales
WHERE (price * .9) > 20
```

In contrast, the following example illustrates the opposite use of an expression, in which the expression is the value you are comparing:

```
SELECT ord_num, ord_date
FROM sales
WHERE (ord_date >= DATEADD(day, -10, GETDATE() ) )
```

The way you specify an expression for searching depends on whether it appears as a condition or as a value to search.

[Use an expression as a condition](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Grid** pane, insert a new grid row.
3. In the **Column** column of the new grid row, type the expression you want to use as the condition.
4. In the **Criteria** column for the new row, type the value to compare against the condition.

Column	Alias	Table	Output	Criteria
price * .9				> 20

[Use an expression as a search value](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. If it is not already in the **Grid** pane, add the data column or expression you want to search.
3. In the **Criteria** column for that data column or expression, enter the expression to use as a search value.

Column	Alias	Table	Output	Criteria
ord_date		sales		>= DATEADD (day, - 10, GETDATE ())

[Show All](#)

Where to enter expressions in queries and filters (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can use [expressions](#) to specify criteria that identify the records you want to work with in:

- The [design grid](#) of a query.
- The design grid of the [Advanced Filter/Sort window](#).
- An [SQL statement](#) in a query's [SQL view](#).
- The [Filter By Form](#) window.

In queries, you can also use expressions to update values in a group of records or to create [calculated fields](#).

You can see how Microsoft Access evaluates query expressions by switching to a query's SQL view to display its underlying SQL statements. If your query includes linked tables, the values you specify in criteria on fields from the linked tables are [case-sensitive](#)— they must match the case of the values in the underlying table.

 [Where to enter expressions](#)

In the design grid, you enter the expression in different places depending on the operation you're trying to perform. After you type an expression in the design grid and press the ENTER key, Access displays the expression using syntax it understands. If you don't include an [operator](#) with the expression, Access assumes you want the = operator. For example, if you type **Denmark** as the criteria in a Country field, Access displays "Denmark", and interprets the expression as Country="Denmark". If you include references to field names in the expression, the fields must be in one of the tables added to the query (unless

you are using the **DLookup** function or a subquery). You can't drag the fields to the expression in the design grid, however. The following table shows where you can enter expressions.

To do this	Type the expression in
Specify criteria for a select , crosstab , or action query , or for an advanced filter	A Criteria cell in the design grid of a query or advanced filter.
Create a calculated field	A Field cell in the design grid of a query or advanced filter.
Update records according to the results of the expression (update queries only)	An Update To cell in the design grid of an update query.
Specify the records you want to work with, how they are grouped, the criteria for the groupings, or how records are sorted	An SQL statement in SQL view.
Specify criteria when using Filter By Form	A field in the Filter By Form window for a custom expression, or select an expression from the list in the field.

[Show All](#)

About calculations in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

There are many types of calculations you can perform in a query. For example, you can calculate the sum or average of the values in one field, multiply the values in two fields, or calculate the date three months from the current date.

You perform calculations in a query using:

- Predefined calculations, called "totals," to compute the following amounts for groups of records or for all the records combined in the query: sum, average, count, minimum, maximum, standard deviation, or variance.
- A custom calculation to perform numeric, date, and text calculations on each record using data from one or more fields. You need to create a new [calculated field](#) directly in the [design grid](#) for these types of calculations.

When you display the results of a calculation in a field, the results aren't actually stored in the underlying table. Instead, Microsoft Access reruns the calculation each time you run the query so that the results are always based on the most current data in the database. Therefore, you can't manually update the calculated results.

[Predefined calculations that use aggregate functions](#)

To display the results of a calculation in a field, you can use a predefined calculation that Access provides or custom calculations you define. Use the predefined calculations, called [aggregate functions](#) or "totals," if you want to compute the following amounts for all records or for groups of records: sum, average, count, minimum, maximum, standard deviation, or variance. You choose one totals calculation for each field you want to calculate.

1

CountOfOrderID	SumOfExtPrice
2820	4835407.19

2

Category Name	CountOfProductID
Beverages	12
Condiments	12

- 1 Calculations on all records
- 2 Calculations on groups of records

You can calculate some types of totals using the Simple Query Wizard. Or, you can calculate all types of totals using the **Total** row in the query design grid, where you select the aggregate function for the calculation you want to perform on a field.

In the query design grid, you can also specify criteria to affect the calculations and produce different query results. By adding criteria, you can limit the:

- Groups before performing calculations on those groups.
- Results after calculations on the groups are performed.
- Records before they are grouped and before calculations are performed.

[About aggregate functions and other options in the Total row of the query design grid](#)

[Aggregate functions](#)

The following options in the query design grid's **Total** row are [aggregate functions](#):

Select	To find the	Use with these field data types
Sum	Total of the values in a field.	Number, Date/Time, Currency, and AutoNumber
Avg	Average of the values in a field.	Number, Date/Time, Currency, and AutoNumber
Min	Lowest value in a field.	Text, Number, Date/Time, Currency, and AutoNumber
Max	Highest value in a field.	Text, Number, Date/Time, Currency, and AutoNumber
Count	Number of values in a field, not counting Null (blank)	Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, and

	values.	OLE Object
StDev	Standard deviation of the values in a field.	Number, Date/Time, Currency, and AutoNumber
Var	Variance of the values in a field.	Number, Date/Time, Currency, and AutoNumber

[First and Last functions](#)

You use the **First** or **Last** functions to return the first or last record in the group you are performing calculations on. These functions return the first or last record as it was entered in chronological order. Sorting the records has no effect on these functions.

[Group By, Expression, and Where options](#)

The following table summarizes what each option does:

Select	To
Group By	Define the groups you want to perform the calculations for. For example, to show total sales by category, select Group By for the CategoryName field.
Expression	Create a calculated field that includes an aggregate function in its expression . Usually, you create a calculated field when you want to use multiple functions in an expression.
Where	Specify criteria for a field you aren't using to define groupings. If you select this option for a field, Access will hide the field in the query results by clearing the Show check box.

[Custom calculations and calculated fields](#)

With a custom calculation, you can perform numeric, date, and text calculations on each record using data from one or more fields. For example, with a custom calculation, you can multiply one field's values by a set amount, find the difference between two dates stored in separate fields, combine several values in a Text field, or create [subqueries](#). Using the options in the **Total** row in the design grid, you can perform the calculation on groups of records and calculate a sum, average, count, or other type of total on the calculated field.

For custom calculations, you need to create a new calculated field directly in the design grid. You create a calculated field by entering an [expression](#) into an empty **Field** cell in the query design grid.

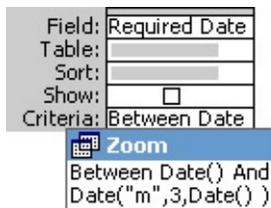
Field: OrderAmt: [Quantity]*[UnitPrice]

The expression can be made up of multiple calculations as the following example shows:

Sum([UnitsInStock]+[UnitsOnOrder])

You can also specify criteria for a calculated field to affect the results of the calculation.

You don't have to display the results of a calculation in a field, however. Instead, you can use them as criteria to determine the records the query selects or to determine which records to perform an action on. For example, you can specify the following expression in the **Criteria** row to tell the query to return only records that have values in the RequiredDate field that are between today's date and three months from today's date.



[Updating data with calculations](#)

You can also use a calculation to update data from an update query. For example, you can enter the following [expression](#) in the **Update To** cell to increase all the values in the UnitPrice field by 5 percent.

Field: UnitPrice
Table:
Update To: [UnitPrice]*1.05

[Handling Null values in calculations](#)

The [aggregate functions](#) won't include records containing blank (**Null**) values in their calculations. For example, the **Count** function returns a count of all the

records without **Null** values. If you want to find the total number of records including those with **Null** values, use **Count** with the asterisk (*) wildcard character.

Count(*)

To count **Null** values when using the other aggregate functions, use the **Nz** function, which converts **Null** values to zeroes so they are included in a calculation.

FreightPercentage: Sum(Nz([Freight], 0)) / Sum(Nz([Subtotal]), 0) *

When you use an arithmetic [operator](#) (+, -, *, /) in an [expression](#) and the value of one of the fields in the expression is **Null**, the result of the entire expression will be **Null**. If some records in one of the fields you used in the expression might have a **Null** value, you can convert the **Null** value to zero using the **Nz** function as shown in the following example:

Nz([UnitsInStock], 0) + Nz([UnitsOnOrder], 0)

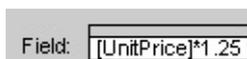
[Show All](#)

Create a calculated field in a query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

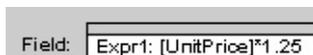
You can create a new field that displays the results of a calculation you define with an [expression](#) or that manipulates field values.

1. Open the query in [Design view](#).
2. Type an expression in an empty cell in the **Field** row. If the expression includes a field name, you must place brackets around the name.



Field:	[UnitPrice]*1.25
--------	------------------

After you press ENTER or move to another cell, Microsoft Access enters the default field name ExprN, where N is an integer incremented for each new expression field in the query. The name appears before the expression and is followed by a colon. In a [datasheet](#), this name is the column heading.



Field:	Expr1: [UnitPrice]*1.25
--------	-------------------------

You can select ExprN and type a more descriptive name, such as NewPrice.

3. If the expression includes one or more aggregate functions (**Sum**, **Avg**, **Count**, **Min**, **Max**, **StDev**, or **Var**), and if the [design grid](#) includes another field or fields that you want to use for grouping, click **Totals** Σ on the toolbar (unless the **Total** row is already displayed). Leave **Group By** in the **Total** cell for the grouping field, and in the [calculated field](#), change **Group By** to **Expression**.



Field:	CustomerID	FreightPercentage: Sum
Table:	Orders	
Total:	Group By	Expression

Zoom
Sum([Freight])/Sum([Subtotal])*100

4. If you want, enter criteria to affect results of the calculation.

[How?](#)

In the query [design grid](#), where you specify criteria determines when the calculation is performed and can produce different query results.

Note Although the following examples only show queries that calculate totals, these guidelines apply to [calculated fields](#) as well.

[Limit groups before performing calculations on groups of records](#)

- Specify the criteria in the **Group By** fields as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum 1
Sort:			
Show:			
Criteria:	"Canada"		
Or:	"UK"		

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Paillarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

- 1** This query totals extended prices for ...
- 2** ... companies in Canada and the UK only.

[Return selected results after calculations are performed](#)

- Specify criteria in the field that contains the calculation as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum
Sort:			
Show:			
Criteria:	"Canada"		<10000
or:	"UK"		<10000

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Pailarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$1,383.90

Ship Country	Company Name	SumOfExtendedPrice
Canada	Laughing Bacchus	\$522.50
UK	B's Beverages	\$1,383.90

❶ This query totals extended prices for companies in Canada and the UK ...

❷ ... but shows only those that are less than \$10,000.

[Limit records before they are grouped and before the calculation is performed](#)

- Add to the design grid the field whose records you want to limit, and then specify criteria in the field's **Criteria** cell.

If you're calculating totals in the same query, set the **Total** cell for the field containing the criteria to **Where**. This rule applies whether you're calculating the total on all records or groups of records. (Microsoft Access automatically clears the **Show** check box.)

The following example uses the ExtendedPrice field twice, once to limit the records and once to calculate the total. However, you can use a different field to limit records by dragging that field to the design grid and setting its **Total** cell to **Where**.

Ship Country	Company Name	ExtendedPrice
UK	Seven Seas Imports	\$345.00
Germany	Lehmanns Marktstand	\$525.00
UK	Seven Seas Imports	\$210.00
Canada	Mère Paillarde	\$691.20

1

Field:	ShipCountry	Company Name	ExtendedPrice	ExtendedPrice
Table:				
Total:	Group By	Group By	Sum	Where
Sort:				
Show:				
Criteria:	"Canada"			<500
or:	"UK"			

2

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$11,938.00
Canada	Mère Paillarde	\$21,691.36
UK	Around the Horn	\$4,907.50
UK	B's Beverages	\$1,380.00
UK	Eastern Connection	\$11,463.00
UK	Seven Seas Imports	\$9799.60

3

- 1 From these records, the query retrieves only those with extended prices greater than \$500.00 before it groups or totals ...
- 2 ... and then it sums and shows only those totals for companies in Canada or the UK.
- 3 This total for Seven Seas Imports does not include the order for \$210.00.

Notes

- The Order Details Extended query in the Northwind sample database has an example of a calculated field called ExtendedPrice. The Order Subtotals query has an example of a calculated field called Subtotal that calculates on groups of records. To view these queries, open the Northwind database in your Office folder's Samples subfolder and then open the Order Details Extended or Order Subtotals query in Design view.
- You might want to set field properties, such as the **Format** property, since the field doesn't inherit properties from the underlying table.

[Show All](#)

Total records in a query (MDB)

[Calculate totals on all the records in a query](#)

1. Create a [select query](#) in [Design view](#) and add the tables whose records you want to use in the calculation.
2. Add the fields on which you want to perform calculations and specify [criteria](#).
3. Click **Totals**  on the toolbar.

Microsoft Access displays the **Total** row in the [design grid](#).

4. For each field in the design grid, click its cell in the **Total** row, and then click one of the following aggregate functions: **Sum**, **Avg**, **Min**, **Max**, **Count**, **StDev**, or **Var**, or click the other functions, **First** and **Last**.
5. If you want, enter criteria to affect the results of the calculation.

[How?](#)

In the query [design grid](#), where you specify criteria determines when the calculation is performed and can produce different query results.

Note Although the following examples only show queries that calculate totals, these guidelines apply to [calculated fields](#) as well.

[Limit groups before performing calculations on groups of records](#)

- Specify the criteria in the **Group By** fields as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum 1
Sort:			
Show:			
Criteria:	"Canada"		
Or:	"UK"		

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Pailarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

- 1 This query totals extended prices for ...
- 2 ... companies in Canada and the UK only.

[Return selected results after calculations are performed](#)

- o Specify criteria in the field that contains the calculation as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum
Sort:			
Show:			
Criteria:	"Canada"		<10000
Or:	"UK"		<10000

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Pailarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

Ship Country	Company Name	SumOfExtendedPrice
Canada	Laughing Bacchus	\$522.50
UK	B's Beverages	\$7,383.90

- 1 This query totals extended prices for companies in Canada and the UK ...
- 2 ... but shows only those that are less than \$10,000.

[Limit records before they are grouped and before the calculation is performed](#)

- o Add to the design grid the field whose records you want to limit, and then specify criteria in the field's **Criteria** cell.

If you're calculating totals in the same query, set the **Total** cell for the field containing the criteria to **Where**. This rule applies whether you're

calculating the total on all records or groups of records. (Microsoft Access automatically clears the **Show** check box.)

The following example uses the ExtendedPrice field twice, once to limit the records and once to calculate the total. However, you can use a different field to limit records by dragging that field to the design grid and setting its **Total** cell to **Where**.

Ship Country	Company Name	ExtendedPrice
UK	Seven Seas Imports	\$345.00
Germany	Lehmanns Marktstand	\$525.00
UK	Seven Seas Imports	\$210.00
Canada	Mère Paillarde	\$631.20

1

Field:	ShipCountry	Company Name	ExtendedPrice	ExtendedPrice
Table:				
Total:	Group By	Group By	Sum	Where
Sort:				
Show:				
Criteria:	"Canada"			<500
or:	"UK"			

2

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$11,338.00
Canada	Mère Paillarde	\$21,631.36
UK	Around the Horn	\$4,307.50
UK	B's Beverages	\$1,380.00
UK	Eastern Connection	\$11,463.00
UK	Seven Seas Imports	\$9739.60

3

1 From these records, the query retrieves only those with extended prices greater than \$500.00 before it groups or totals ...

2 ... and then it sums and shows only those totals for companies in Canada or the UK.

3 This total for Seven Seas Imports does not include the order for \$210.00.

6. If you want, sort the results.
7. Click **View**  on the toolbar to view the results.

[Calculate totals on groups of records in a query](#)

1. Create a [select query](#) in [Design view](#) and add the tables whose records you want to use in the calculation.
2. Add the fields on which you want to perform calculations, define groupings, and specify [criteria](#).
3. Click **Totals**  on the toolbar.

Microsoft Access displays the **Total** row in the [design grid](#).

4. For the field or fields you want to group on, leave **Group By** in the **Total** cell.
5. For each field you want to calculate, click its cell in the **Total** row, and then click one of the following aggregate functions: **Sum**, **Avg**, **Min**, **Max**, **Count**, **StDev**, or **Var**, or click the other functions, **First** and **Last**.
6. If you want, enter criteria to affect the results of the calculation.

[How?](#)

In the query [design grid](#), where you specify criteria determines when the calculation is performed and can produce different query results.

Note Although the following examples only show queries that calculate totals, these guidelines apply to [calculated fields](#) as well.

[Limit groups before performing calculations on groups of records](#)

- o Specify the criteria in the **Group By** fields as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum 1
Sort:			
Show:			
Criteria:	"Canada"		
Or:	"UK"		

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Paillede	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$7,383.90

- 1** This query totals extended prices for ...
- 2** ... companies in Canada and the UK only.

[Return selected results after calculations are performed](#)

- o Specify criteria in the field that contains the calculation as the following example shows.

Field:	ShipCountry	Company Name	ExtendedPrice
Table:			
Total:	Group By	Group By	Sum
Sort:			
Show:			
Criteria:	"Canada"		<10000
or:	"UK"		<10000

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$28,025.51
Canada	Laughing Bacchus	\$522.50
Canada	Mère Pailarde	\$37,123.65
UK	Around the Horn	\$14,602.15
UK	B's Beverages	\$1,383.90

Ship Country	Company Name	SumOfExtendedPrice
Canada	Laughing Bacchus	\$522.50
UK	B's Beverages	\$1,383.90

❶ This query totals extended prices for companies in Canada and the UK ...

❷ ... but shows only those that are less than \$10,000.

[Limit records before they are grouped and before the calculation is performed](#)

- Add to the design grid the field whose records you want to limit, and then specify criteria in the field's **Criteria** cell.

If you're calculating totals in the same query, set the **Total** cell for the field containing the criteria to **Where**. This rule applies whether you're calculating the total on all records or groups of records. (Microsoft Access automatically clears the **Show** check box.)

The following example uses the ExtendedPrice field twice, once to limit the records and once to calculate the total. However, you can use a different field to limit records by dragging that field to the design grid and setting its **Total** cell to **Where**.

Ship Country	Company Name	ExtendedPrice
UK	Seven Seas Imports	\$345.00
Germany	Lehmanns Marktstand	\$525.00
UK	Seven Seas Imports	\$210.00
Canada	Mère Paillarde	\$691.20

1

Field:	ShipCountry	Company Name	ExtendedPrice	ExtendedPrice
Table:				
Total:	Group By	Group By	Sum	Where
Sort:				
Show:				
Criteria:	"Canada"			<500
or:	"UK"			

2

Ship Country	Company Name	SumOfExtendedPrice
Canada	Bottom-Dollar Markets	\$11,938.00
Canada	Mère Paillarde	\$21,691.36
UK	Around the Horn	\$4,907.50
UK	B's Beverages	\$1,380.00
UK	Eastern Connection	\$11,463.00
UK	Seven Seas Imports	\$9799.60

3

- 1 From these records, the query retrieves only those with extended prices greater than \$500.00 before it groups or totals ...
- 2 ... and then it sums and shows only those totals for companies in Canada or the UK.
- 3 This total for Seven Seas Imports does not include the order for \$210.00.

7. If you want, sort the results.
8. Click **View**  on the toolbar to view the results.

Note If you add a [calculated field](#) that includes one or more [aggregate functions](#) to a query in which you're calculating totals on all the records, you must set the calculated field's **Total** cell to **Expression**.

[Show All](#)

About SQL queries (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

An SQL query is a query you create by using an [SQL statement](#). You can use Structured Query Language (SQL) to query, update, and manage relational databases such as Microsoft Access.

When you create a query in query [Design view](#), Access constructs the equivalent SQL statements behind the scenes for you. In fact, most query properties in the property sheet in query Design view have equivalent clauses and options available in [SQL view](#). If you want, you can view or edit the SQL statement in SQL view. However, after you make changes to a query in SQL view, the query might not be displayed the way it was previously in Design view.

Some SQL queries, called [SQL-specific queries](#), can't be created in the [design grid](#). For [pass-through](#), [data-definition](#), and [union queries](#), you must create the SQL statements directly in SQL view. For [subqueries](#), you enter the SQL in the **Field** row or the **Criteria** row of the query design grid.

[Where SQL statements are used](#)

You can use SQL statements in many places in Access where you can enter the name of a table, query, or field.

In some cases, Access fills in the SQL statement for you. For example, when you use a wizard to create a form or report that gets data from more than one table, Access automatically creates an SQL statement that it uses as the setting for the **RecordSource** property of the form or report. When you create a list box or combo box with a wizard, Access creates an SQL statement and uses it as the setting for the **RowSource** property of the list box or combo box.

Without using a wizard, you can generate an SQL statement for the **RecordSource** or **RowSource** properties by clicking the **Build** button next to either of these properties, and then creating a query in query Design view.

You can also use SQL statements programmatically in:

- The SQL Statement argument of the RunSQL macro action.
- Code as a literal string, or as an SQL statement that includes variables and [controls](#).
- The **SQL** property of a **QueryDef** object to change the underlying SQL statement of a query.

[Using expressions in SQL](#)

You can type an expression in an SQL SELECT statement, or in WHERE, ORDER BY, GROUP BY, or HAVING clauses. You can also type an SQL expression in several [arguments](#) and [property](#) settings. For example, you can use an SQL expression as a:

- Where Condition argument of the OpenForm or ApplyFilter action.
- Domain or criteria argument in a [domain aggregate function](#).
- Setting for the **RecordSource** or **RowSource** property in forms and reports.

[SQL-specific queries](#)

[Union query](#)

A union query combines fields from two or more tables or queries into one field in the query's results. You might use a union query to combine data from two tables. For example, you might combine company name and city data for all companies in Brazil that are listed in the Suppliers and Customers tables. You could then create a make-table query based on the union query to make a new table.



- 1 The first SELECT statement returns two fields ...
- 2 ... the second SELECT statement returns two corresponding fields, and then ...
- 3 ... combines the values in the corresponding fields from both tables into one field.

[Pass-through query](#)

A pass-through query sends commands directly to [ODBC databases](#), such as Microsoft FoxPro, using commands that are accepted by the server. For example, you can use a pass-through query to retrieve records or change data. With pass-through queries, you work directly with the tables on the server instead of linking to them. Pass-through queries are also useful for running [stored procedures](#) on an [ODBC](#) server.

[Data-definition query](#)

A data-definition query creates, deletes, or alters tables, or creates indexes in a database table. For example, the following data-definition query uses the CREATE TABLE statement to create a table named Friends. The statement includes the name and [data type](#) for each field in the table and assigns the FriendID field an [index](#) that marks it as the [primary key](#).

```
CREATE TABLE Friends
([FriendID] integer,
[LastName] text,
[FirstName] text,
[Birthdate] date,
[Phone] text,
[Notes] memo,
CONSTRAINT [Index1] PRIMARY KEY ([FriendID]));
```

[Subquery](#)

A subquery consists of an SQL SELECT statement inside another select query or action query. You can enter these statements in the **Field** row of the query [design grid](#) to define a new field, or in the **Criteria** row to define criteria for a field.

You can use subqueries to:

- Test for a result from the subquery (by using the EXISTS or NOT EXISTS reserved words).
- Find any values in the main query that are equal to, greater than, or less than values returned by the subquery (by using the ANY, IN, or ALL reserved words).
- Create subqueries within subqueries (nested subqueries).

[Show All](#)

Create an SQL-specific query (MDB)

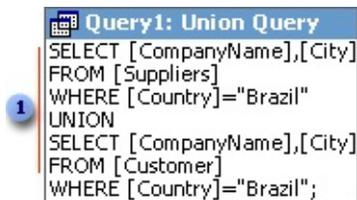
Note The information in this topic applies only to a Microsoft Access database (.mdb).

Caution If you convert a [union query](#), [pass-through query](#), or [data-definition query](#) to another type of query, such as a [select query](#), you'll lose the [SQL statement](#) that you enter.

 [Create a union query that combines corresponding fields from two or more tables or queries into one field](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. Without adding tables or queries, click **Close** in the **Show Table** dialog box.
4. On the **Query** menu, point to **SQL Specific**, and then click **Union**.
5. Enter SQL SELECT statements combined with either the UNION operation if you don't want to return duplicate records or the UNION ALL operation if you do want to return duplicate records.

Note Each SELECT statement must return the same number of fields, in the same order. The corresponding fields are required to have compatible [data types](#) with one exception: You can use a Number and a Text field as corresponding fields.



```
Query1: Union Query
SELECT [CompanyName],[City]
FROM [Suppliers]
WHERE [Country]="Brazil"
UNION
SELECT [CompanyName],[City]
FROM [Customer]
WHERE [Country]="Brazil";
```

1 This union query combines the values in the CompanyName and City fields from the Suppliers and Customers tables if the Country field is "Brazil".

6. If you want to specify sorting in a union query, add a single ORDER BY

clause to the end of the last SELECT statement. In the ORDER BY clause, specify the field name to sort, which must come from the first SELECT statement.

7. To see the query's results, click **View**  on the toolbar.

Note A union query takes its column names from the column names in the first table or SELECT statement. If you want to rename a field in the results, use the AS clause to create an alias for the fields.

[Create a pass-through query that sends commands to an SQL database](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. Without adding tables or queries, click **Close** in the **Show Table** dialog box.
4. On the **Query** menu, point to **SQL Specific**, and then click **Pass-Through**.
5. On the toolbar, click **Properties**  to display the query property sheet.
6. In the query property sheet, set the **ODBCConnectStr** property to specify information on the database to which you want to connect. You can type the connection information, or click **Build** , and then enter information about the server you're connecting to.

When you are prompted to save the password in the connection string, select **Yes** if you want the password and logon stored in the connection string information.

Security Avoid saving logon information when connecting to data sources. This information may be stored as plain text, and a malicious user could access the information to compromise the security of the data source.

7. If the query isn't the type that returns records, set the **ReturnsRecords** property to **No**.
8. In the SQL Pass-Through Query window, type your [pass-through query](#).

For details on the syntax for your query, see the documentation for the SQL database server to which you're sending the query.

9. To run the query, click **Run**  on the toolbar. (For a pass-through query that returns records, you can click **View**  on the toolbar, instead.)

If necessary, Microsoft Access prompts you for information about your SQL server database.

Notes

- If you don't specify a connection string in the **ODBCConnectStr** property, or if you delete an existing string, Access uses the default string "ODBC;". With this setting, Access prompts you for connection information each time you run the query.
- Some pass-through queries can return messages in addition to data. If you set the query's **LogMessages** property to **Yes**, Access creates a table that contains any returned messages. The table name is the user name concatenated with a hyphen (-) and a sequential number starting at 00. For example, the default user name is ADMIN so the tables returned would be named "ADMIN - 00," "ADMIN - 01," and so on.

 [Create a data-definition query to create, delete, or alter tables, or to create indexes in a database](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. Without adding tables or queries, click **Close** in the **Show Table** dialog box.
4. On the **Query** menu, point to **SQL Specific**, and then click **Data Definition**.
5. Enter the SQL statement for your data-definition query. Each data-definition query consists of just one data-definition statement.

 [Supported data-definition statements](#)

SQL statement	Purpose
CREATE TABLE	Creates a table
ALTER TABLE	Adds a new field or constraint to an existing table
DROP	Deletes a table from a database or removes an index from a field or group of fields

CREATE INDEX Creates an index for a field or group of fields

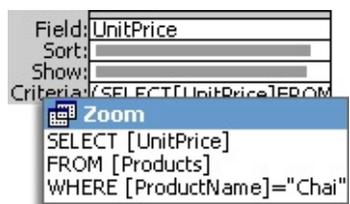
To run the query, click **Run**  on the toolbar.

 [Create a subquery to define a field or define criteria for a field](#)

1. Create a new query.

 [How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
 2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
 3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
 4. Double-click the name of each object you want to add to the query, and then click **Close**.
 5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
 6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), add the fields you want to the [design grid](#), including the fields you want to use the [subquery](#) for.
 3. If you're using a subquery to define criteria for a field, type a SELECT statement enclosed in parentheses in the **Criteria** cell of the field for which you want to set criteria.



If you're using a subquery to define a **Field** cell, type a SELECT statement enclosed in parentheses in a **Field** cell. After you leave the cell, Microsoft Access automatically inserts "Expr1:", "Expr2:", and so on, before the SELECT statement.



To rename the field, replace "Expr1:" with a field name— for example, "Cat:".

Note For more space in which to enter the SELECT statement while in a **Field** or **Criteria** cell, press SHIFT+F2 and enter the statement in the **Zoom** box.

4. To see the results, click **View**  on the toolbar.

Notes

- You can't use a field that you define with a subquery to group records.
- A subquery's SELECT statement can't define a union or [crosstab query](#).

[Show All](#)

View, modify, or copy a query's SQL statement (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

If you're familiar with [SQL](#), you might want to work directly with a query's underlying [SQL statement](#). Or, you might want to copy the SQL statement and paste it elsewhere, such as in the **RowSource** property of a list box in a form or report.

1. Open the query you want to work with.
2. Click the arrow next to **View**  on the toolbar, and then click **SQL View**.

Microsoft Access displays the SQL statement that is equivalent to what you created in [Design view](#).

3. Do one or more of the following:

To modify the SQL, type your changes in the SQL statement.

To copy the SQL, select all or part of the SQL statement, copy the selected text, and paste the SQL statement in, for example, a property box or the Microsoft Visual Basic Editor.

4. To see the changes in the [design grid](#), click the arrow next to the **View** button on the toolbar, and then click **Design View**.

[Show All](#)

Create tables from the results of a pass-through query (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Create a [pass-through query](#).

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. Without adding tables or queries, click **Close** in the **Show Table** dialog box.
4. On the **Query** menu, point to **SQL Specific**, and then click **Pass-Through**.
5. On the toolbar, click **Properties**  to display the query property sheet.
6. In the query property sheet, set the **ODBCConnectStr** property to specify information on the database to which you want to connect. You can type the connection information, or click **Build** , and then enter information about the server you're connecting to.

When you are prompted to save the password in the connection string, select **Yes** if you want the password and logon stored in the connection string information.

7. If the query isn't the type that returns records, set the **ReturnsRecords** property to **No**.
8. In the SQL Pass-Through Query window, type your [pass-through query](#).

For details on the syntax for your query, see the documentation for the SQL database server to which you're sending the query.

9. To run the query, click **Run**  on the toolbar. (For a pass-through query that returns records, you can click **View**  on the toolbar, instead.)

If necessary, Microsoft Access prompts you for information about your SQL server database.

Notes

- If you don't specify a connection string in the **ODBCConnectStr** property, or if you delete an existing string, Access uses the default string "ODBC;". With this setting, Access prompts you for connection information each time you run the query.
 - Some pass-through queries can return messages in addition to data. If you set the query's **LogMessages** property to **Yes**, Access creates a table that contains any returned messages. The table name is the user name concatenated with a hyphen (-) and a sequential number starting at 00. For example, the default user name is ADMIN so the tables returned would be named "ADMIN - 00," "ADMIN - 01," and so on.
2. Create a [make-table query](#) based on the pass-through query.

[How?](#)

1. Create a query, selecting the tables or queries that contain the records you want to put in the new table.

[How?](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.

2. In query [Design view](#), click the arrow next to **Query Type**  on the [toolbar](#), and then click **Make Table**. The **Make Table** dialog box appears.
3. In the **Table Name** box, enter the name of the table you want to create or replace.
4. Do one of the following:

Click **Current Database**.

Click **Another Database**, and then either type the path of the database where you want to put the new table or click **Browse** to locate the database.

5. Click **OK**.
6. Drag from the [field list](#) to the query [design grid](#) the fields you want in the new table.
7. In the **Criteria** cell for the fields that you've dragged to the grid, type the [criteria](#).
8. To preview the new table before you create it, click **View**  on the toolbar. To return to query Design view and make changes or run the query, click **View**  on the toolbar again.
9. To create the new table, click **Run**  on the toolbar.

Note The data in the new table you create does not inherit the field properties or the [primary key](#) setting from the original table.

3. In the make-table query, include all fields from the pass-through query by dragging the asterisk (*) to the [design grid](#).
4. Run the make-table query.

[How?](#)

Do one of the following:

[Run a select or crosstab query](#)

When you open a [select](#) or [crosstab query](#), Microsoft Access runs (executes) the query for you and shows the results in [Datasheet view](#).

1. In the [Database window](#), click **Queries**  under **Objects**.
2. Click the query you want to open.
3. Click **Open** on the Database window toolbar.

[Run an action query](#)

Unlike [select](#) and [crosstab queries](#), you can't view the results of an [action query](#) by opening it in [Datasheet view](#). However, in Datasheet view you can preview the data that will be affected when you run the action query.

Caution It's a good idea to make a copy of the data you are changing or moving in an action query, in case you need to restore the data to its original state after running the action query.

1. Open the action query in [Design view](#).
2. To preview the records that will be affected in Datasheet view, click **View**  on the toolbar and check the records. For each action query, you will see the following:

For this query	The datasheet displays
Update	The fields to be updated.
Delete	The records to be deleted.
Make-table	The fields to be included in the new table.
Append	The records to be added to another table.

3. To return to query Design view, click **View**  on the toolbar again. Make any changes you want in Design view.
4. Click **Run**  on the toolbar to run the query.

Note To stop a query after you start it, press CTRL+BREAK.

When you run the make-table query, you get one table for each result. The first table created has the name defined in the query; subsequent tables have that name concatenated with a sequential number starting at 1.

[Show All](#)

Examples of union queries (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Sorting records](#)

The following [union query](#) selects all company names and cities from both the Suppliers and Customers tables, and sorts the data alphabetically by city.

```
SELECT [CompanyName], [City]
FROM [Suppliers]

UNION SELECT [CompanyName], [City]
FROM [Customers]
ORDER BY [City];
```

[Renaming fields](#)

The following [union query](#) renames the CompanyName field to "Supplier/Customer Name" in the query output.

```
SELECT [CompanyName] AS [Supplier/Customer Name], [City]
FROM [Suppliers]

UNION SELECT [CompanyName] AS [Supplier/Customer Name], [City]
FROM [Customers];
```

[Returning duplicate records](#)

The following [union query](#) uses the UNION ALL statement to retrieve all records, including duplicates.

```
SELECT [CompanyName], [City]
FROM [Suppliers]

UNION ALL SELECT [CompanyName], [City]
FROM [Customers];
```


[Show All](#)

Examples of data-definition queries (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Creating a table](#)

This [data-definition query](#) uses the CREATE TABLE statement to create a table named Friends. The statement includes the name and [data type](#) for each field in the table and assigns the FriendID field an [index](#) that marks it as the [primary key](#).

```
CREATE TABLE Friends
([FriendID] integer,
[LastName] text,
[FirstName] text,
[Birthdate] date,
[Phone] text,
[Notes] memo,
CONSTRAINT [Index1] PRIMARY KEY ([FriendID]));
```

[Creating an index](#)

This [data-definition query](#) uses the CREATE INDEX statement to create a multiple-field [index](#) on the LastName and FirstName fields of a table called Friends.

```
CREATE INDEX NewIndex
ON Friends ([LastName], [FirstName]);
```

[Show All](#)

About ANSI SQL query mode (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

You can create SQL queries in one of two [ANSI SQL query modes](#):

- ANSI-89 describes the traditional [Jet](#) SQL syntax. This mode conforms closely to the ANSI-89 Level 1 specification, but is not ANSI-89 Level 1 compliant. Certain ANSI-89 SQL features are not implemented and the wildcard characters conform to the Visual Basic for Applications (VBA) specification, not SQL.
- ANSI-92 provides new reserved words, syntax rules, and wildcard characters that enhance your ability to create queries, filters, and SQL statements. This mode conforms closely to the ANSI-92 Level 1 specification, but is not ANSI-92 Level 1 compliant. This query mode has more of the ANSI syntax, and the wildcard characters conform to the SQL specification.

In Microsoft Access 2000 using [ADOX](#), you could programmatically create queries that used ANSI-92 SQL syntax. However, any queries you created were not visible in the [Database window](#) because there was no option to set this mode in the user interface. Now in Access 2002 or later, you can set the ANSI SQL query mode through the user interface for the current database and as the default setting for new databases.

[Why use ANSI-92 SQL?](#)

You may want to use ANSI-92 SQL for the following reasons:

- You anticipate upsizing your application in the future to an [Access project](#) and want to create queries that will run with minimal changes in a [Microsoft SQL Server database](#).
- You want to take advantage of the new features not found in ANSI-89 SQL, such as:

- Changing security settings by using the GRANT and REVOKE SQL statements
- Using DISTINCT in an aggregate function reference, for example, SUM(DISTINCT Price). This only applies to Access projects (.adp). It does not apply to Access database (.mdb) files.
- Using the LIMIT TO nn ROWS clause to limit the number of rows returned by a query

[Why you should avoid mixing queries created under different ANSI SQL query modes](#)

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.

[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS  
[New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should help prevent accidental or intentional changes to the ANSI SQL query mode of your application by helping protect your code and helping prevent the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

[The ANSI SQL query mode default for new Access databases in 2002 - 2003 and 2000 file formats](#)

- ANSI-89 is the default setting for a new Access database in 2002 - 2003 and 2000 file format.
- You cannot set the SQL query mode new database default to ANSI-92 in 2000 file format because the option is disabled; ANSI-89 is the only query mode setting available for a database in Access 2000 file format.

[Show All](#)

About avoiding the mixing of queries created under different ANSI SQL query modes (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

The two [ANSI SQL query modes](#), ANSI-89 and ANSI-92, are not compatible. When you create a [Microsoft Access database](#), you need to decide which query mode you are going to use, because mixing [queries](#) created in both query modes could produce runtime errors or unexpected results. The range of data types, reserved words, and wildcard characters are different in each query mode.



[Example of a query using wildcard characters](#)

A query that uses wildcard characters in a criteria expression can produce different results under each query mode. For example, if you run:

- An ANSI-89 SQL query in a database set to ANSI-92 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U*'
```

It returns all customers from a country/region named "U*", not all countries/regions beginning with the letter "U", because the asterisk (*) is not a wildcard character in ANSI-92 SQL.

- An ANSI-92 SQL query in a database set to ANSI-89 query mode, such as:

```
SELECT * FROM Customers WHERE Country Like 'U%'
```

It returns all customers from a country/region named "U%", not all countries/regions beginning with the letter "U", because the percent sign (%) is not a wildcard character in ANSI-89 SQL.

[Example of a query with a duplicate field and alias name](#)

If a query uses an alias that is the same as a base column name and you create a calculated field using the ambiguous name, the query will produce different results under each query mode. For example, if you run the following query:

```
SELECT Orders.OrderID AS Freight, Orders.Freight, [Freight]*1.1 AS [New Freight] FROM Orders;
```

Under ANSI-92 SQL, Access calculates the New Freight value using the Freight field, and therefore increases each Freight by 10%. Under ANSI-89 SQL, Access calculates the New Freight value using the Freight alias, and therefore increases each OrderID by 10%.

[How to avoid problems caused by mixing queries under different ANSI SQL query modes in the same database](#)

In general, avoid doing the following to prevent problems caused by mixing queries created under different ANSI SQL query modes:

- Converting an Access database set to ANSI-92 SQL query mode from 2002 - 2003 file format to 2000 or 97 file format. ANSI-92 SQL query mode is only supported in the 2002 - 2003 file format.
- Changing the ANSI SQL query mode for the current database after you've created one or more queries. If you must do this, retest the existing queries to ensure that they still run or produce expected results, and rewrite the queries if necessary.
- Importing queries created under one ANSI SQL query mode into an Access database set to another mode, or exporting queries created under one ANSI SQL query mode to an Access database set to another mode.
- Changing the ANSI SQL query mode in an existing application. This could break your application and require rewriting your application.

In Access 2000, you can only programmatically change the ANSI SQL query mode and any queries created under ANSI-92 mode were hidden in the [Database window](#). In Access 2002 or later, you or a user can change ANSI SQL query mode through the user interface and ANSI-92 queries are no longer hidden in the Database window, so you should help prevent accidental or intentional changes to the ANSI SQL query mode of your

application by helping protect your code and helping prevent the changing of the query mode through the application's user interface.

- Using ambiguous aliases and column names. To avoid confusion, ensure that aliases and column names are always unique in an SQL statement.

Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#)
[N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Z](#)

A

[Back to top](#)

action query

A query that copies or changes data. Action queries include append, delete, make-table, and update queries. Delete and update queries change existing data; append and make-table queries copy existing data. In SQL, these are referred to as DML queries and consist of the keywords DELETE, INSERT, and UPDATE. Microsoft Jet supports the keyword SELECT INTO, which is also considered a DDL query. In contrast, select queries return data records. An SQL pass-through query may also be an action query.

Active Data Objects (ADO)

A programming interface to access and manipulate database objects.

ActiveX object

An object that is exposed to other applications or programming tools through Automation (formerly OLE Automation) interfaces. For example, Microsoft Access can create and manipulate Microsoft ActiveX objects within Microsoft Word or Microsoft Excel.

aggregate function

A function, such as **Sum**, **Count**, **Avg**, and **Max**, that you can use to calculate totals. In writing expressions and in programming, you can use SQL aggregate functions (including the four listed here) and domain aggregate functions to determine various statistics.

alias

Visual Basic: An alternate name you give to an external procedure to avoid conflict with a Microsoft Visual Basic keyword, public variable, constant, or a name not allowed by the Visual Basic naming conventions. Structured Query Language (SQL): An alternate name you give to a field or expression in a SELECT statement, using the optional AS clause. ANSI refers to this as a correlation name.

ANSI Character Set

An 8-bit character set used by Microsoft Windows that allows you to represent up to 256 characters (0 through 255) using your keyboard. The ASCII character set is a subset of the ANSI set.

ANSI SQL-92 mode

This mode allows for an increased SQL syntax that is more compliant with SQL-92 and Microsoft SQL Server. This mode is only available when going through the Microsoft OLE DB Provider for Jet. It cannot be set when going through DAO. New features are primarily in DDL. Support for the ANSI wildcards, which are the percent sign (%) and the underscore character (_), are also part of the ANSI SQL-92 mode.

append query

An action query that adds new records to the end of an existing table or query. Append queries do not return records (rows). SQL syntax would contain the INSERT keyword.

argument

A constant, variable, or expression that supplies additional information to an action, procedure, or method. You use an argument with an operand in an expression to supply additional information required by some action.

array

A variable that contains a finite number of elements that have a common name and data type. Each element of an array is identified by a unique index number. Changes made to one element of an array do not affect the other elements.

ASCII Character Set

American Standard Code for Information Interchange (ASCII) 7-bit character set widely used to represent letters and symbols found on a standard U.S. keyboard. The ASCII character set is the same as the first 128 characters (0 – 127) in the ANSI character set.

asterisk

A character (*) used as a wildcard in SQL LIKE expressions to accept any string of characters. For example, the expression LIKE "*town*" returns records containing the string "town" followed by zero or more characters. When you create a query or enter an SQL statement, use the asterisk to include all the fields that currently exist in the underlying table or query. For example, "SELECT * FROM MyTable" returns all fields (columns) from the table. When issuing a query through the Microsoft OLE DB Provider for Jet, the asterisk is no longer used as a wildcard symbol. Instead, the percent sign (%) is used as the ANSI wildcard symbol. Stored

queries will not be affected, but new queries (views) that are created through the Microsoft OLE DB Provider for Jet will need to use the ANSI wildcards, the percent sign (%) and the underscore character (_), to get the behavior similar to that provided by the asterisk (*) and the question mark (?). Microsoft OLE DB Provider for Jet will translate the asterisk and the question mark as literal characters, and the returning result set will be different from going through the Microsoft Access UI or DAO.

asynchronous query

A type of query in which SQL queries return immediately, even though the results are still pending. This enables an application to continue with other processing while the query is pending completion.

AutoNumber data type

A field attribute that automatically stores a unique number for each record as it is added to a table. An AutoNumber field uses the **Long** data type and has its **Attributes** property set to **dbAutoIncrement**. Numbers generated by an AutoNumber field cannot be modified unless special permission has been granted to the table through either the Grant or Revoke statements in SQL DDL syntax. The AutoNumber field is also known as a Counter or Identity data type.

AutoNumber field

A field data type that automatically stores a unique number for each record as it is added to a table. An AutoNumber field always uses the **Long** data type. Numbers generated by an AutoNumber field cannot be modified unless special permission has been granted to the table through either the Grant or Revoke statements in SQL DDL syntax. The AutoNumber field is also known as a Counter or Identity data type. When using SQL DDL, this data type now supports varying seed and increment values. This differs from previous versions where the seed value was 1 and the increment value was 1.

B

[Back to top](#)

base computer

A computer used in a star topology. A replica is copied from the hub server and placed on the base computer. This replica can be used to make additional replicas for other computers.

base table

A table in a Microsoft Jet database. A table defines the structure of a relational database, and is an object that stores data in records and fields. You can manipulate the structure of a base table by using the DAO objects or data definition (DDL) SQL statements, and you can modify data in a base table by using **Recordset** objects or action queries.

Big Integer data type

A data type that stores a signed, exact numeric value with precision 19 (signed) or 20 (unsigned), scale 0 (signed: $-263 = n = 263-1$; unsigned: $0 = n = 264-1$).

Binary data type

A data type that stores fixed-length binary data. The maximum length is 255 bytes.

bookmark

A property of the Recordset object that contains a binary string identifying the current record. If you assign the Bookmark value to a variable and then move to another record, you can make the earlier record current again by setting the Bookmark property to that string variable.

Boolean data type

A **True/False** or yes/no value. **Boolean** values are usually stored in Bit fields in a Microsoft Jet database; however, some databases do not support this data type directly.

Boolean expression

An expression that evaluates to either **True** or **False**.

Byte data type

A fundamental data type used to hold small positive integer numbers ranging from 0 to 255.

C

[Back to top](#)

Cartesian product

The result of joining two relational tables, producing all possible ordered combinations of rows from the first table with all rows from the second table. Generally, a Cartesian product results from executing an SQL SELECT statement referencing two or more tables in the FROM clause and not including a WHERE or JOIN clause that indicates how the tables are to be joined.

cascade

The process of one action triggering another action. For example, when a cascading update relationship is defined for two (or more) tables, an update to the primary key in the primary table automatically triggers (cascades) changes to the foreign table. Now also definable from SQL DDL.

cascading delete

For relationships that enforce referential integrity between tables, cascading delete is an option that causes the deletion of a record from the primary table to automatically delete all related records in the related foreign table or tables. For example, if you establish a relationship between a Customers (primary) table and an Orders (foreign) table with the cascading delete option enabled, when a record in the Customers table is deleted, all orders associated with that customer would also be deleted in the foreign Orders table. Now also definable from SQL DDL.

cascading NULL

For relationships enforcing referential integrity between tables, cascading NULL is an option that causes a deleted row in the primary table to automatically update the foreign key in all related records in the related foreign table or tables with a NULL value. Now also definable from SQL DDL. For example, if you establish a relationship between a Customers (primary) table and an Orders (foreign) table with the cascading NULL option enabled, when the primary key in a record in the Customers table is deleted, all orders in the foreign Orders table associated with that customer would also be changed to have a NULL value.

cascading update

For relationships enforcing referential integrity between tables, cascading update is an option that causes a change to the primary key in a record in the primary table to automatically update the foreign key in all related records in the related foreign table or tables. For example, if you establish a relationship between a Customers (primary) table and an Orders (foreign) table with the cascading update option enabled, when the primary key in a record in the Customers table is changed, all orders associated with that customer would also be changed in the foreign Orders table. Now also definable from SQL DDL.

case-sensitive

Capable of distinguishing between uppercase and lowercase letters. A case-sensitive search finds only text that is an exact match of uppercase and lowercase letters. Such a search would, for instance, treat "ZeroLengthStr" and "zerolengthstr" as different. Microsoft Jet database operations are not case-sensitive. However, case sensitivity is a feature of some other database

management systems.

Char data type

A data type that stores a fixed-length character string. The length is set by the **Size** property.

CHECK constraint

An ANSI SQL-92 concept that allows for business rules that span an entire table or multiple other tables to be created. An example of this would be to create a Customer table and an Order table. The Order table could have a CHECK constraint that would prevent orders for a customer from exceeding a credit limit defined for the customer in the Customer table.

class

The formal definition of an object. The class acts as the template from which an instance of an object is created at run time. The class defines the properties of the object and the methods used to control the object's behavior.

client batch cursor library

A library that provides client-side cursor support for ODBC database applications. This library supports all four types of cursors (keyset, static, dynamic, and forward-only) and provides a number of other features including the ability to dissociate connections and perform optimistic batch updates.

clustered index

The physical order of rows is the same as the indexed order of rows.

collection

An object that contains a set of related objects. An object's position in the collection can change whenever a change occurs in the collection; therefore, the position of any specific object in the collection may vary.

collision

A conflict that occurs during a batch update. A collision occurs when a client reads data from the server and then attempts to modify that data in a batch update, but before the update attempt is actually executed another client changes the original server data. In this situation, the first client is attempting to modify server data without knowing what data actually exists on the server.

column

The visual representation of a field in a grid. A column defines the data type, size, and other attributes of one field of a row (record) of data. All columns taken as a set define a row (record) in the database. An individual column contains data related in type and purpose throughout the table; that

is, a column's definition does not change from row to row.

column-level tracking

Column-level tracking is a mechanism where each column in a row participates individually in conflict determination and resolution. If more than one replica modifies the same row, but each replica modifies a unique column, no conflict will occur and all changes will be merged.

commit

To accept a pending transaction. If you use transaction processing and begin a transaction using DAO, ADO, or SQL, none of the changes made in the transaction are written to the database until you commit, or accept, the transaction.

compound query

A query that is composed of at least one action query (a query that copies or changes data) and at least one select query (a query that returns a **Recordset** without changing data). In DAO, a compound query is created by putting two or more SQL statements (separated by semicolons) in the **SQL** property of a **QueryDef** object.

conflict table

A table that is generated when a synchronization conflict occurs between two replicas. The conflict table appears only in the replica with rejected changes.

connection string

A string used to define the source of data for an external database. The connection string is usually assigned to the **Connect** property of a **QueryDef**, **TableDef**, **Connection**, or **Database** object or as an argument to the **OpenDatabase** method.

consistent

The state of a multiple-table **Recordset** object that allows you to perform only updates that result in a consistent view of the data. For example, in a **Recordset** that is a join of two or more tables (a one-to-many relationship), a consistent query would not allow you to set the many-side key to a value that is not in the one-side table.

constant

A named item that retains a constant value throughout the execution of a program, as opposed to a variable, whose value can change during execution. Each host application can define its own set of constants. Additional constants may be defined by the user with the **Const** statement. Constants can be used anywhere in your code in place of actual values. A constant may be a string or numeric literal, another constant, or any

combination that includes arithmetic or logical operators except the **Is** statement and exponentiation. For example: Const strdb = "c:\Program Files\Microsoft Office\Access\Samples\Northwind.mdb"

constraint

A limitation of the possible values that users can enter into a field. The ANSI SQL DDL CHECK constraint has been added to Microsoft Jet version 4.X when going through the Microsoft OLE DB Provider for Jet. The CHECK constraint allows for the creation of business rules that span an entire table or multiple other tables.

control character

Any key combination that begins with the CTRL or ESC key.

copy buffer

A location created by the Microsoft Jet database engine for the contents of a record that is open for editing. The **Edit** method copies the current record to the copy buffer; the **Addnew** method clears the buffer for a new record and sets the default values; and the **Update** method saves the data from the copy buffer to the database, replacing the current record or inserting the new record. Any statement that resets or moves the current record pointer discards the copy buffer. For example, using the **MoveNext** method or changing the **Index** property of a table would discard the contents of the copy buffer.

correlation name

The ANSI definition of a name that refers to a column in a table or view (stored query). Also referred to as an alias name.

criteria

A set of limiting conditions, such as = "**Denmark**" (meaning equal to Denmark) or > **30000** , used in creating a query or filter to show a specific set of records.

crosstab query

A query that calculates a sum, average, count, or other type of total on records, and then groups the result by two types of information, one down the left side of a grid (row headings) and the other across the top (column headings). For example, the Quarterly Orders by Product query in the Northwind sample database is a crosstab query.

Currency data type

A data type that is useful for calculations involving money or for fixed-point calculations in which accuracy is extremely important. This data type is used to store numbers with up to 15 digits to the left of the decimal point and 4 digits to the right. Because the **Currency** data type uses discrete

values for all amounts, binary round-off is not a factor when calculating totals.

current database

The **Database** object returned by the **CurrentDB**() function. A reference of **DBEngine.Workspaces(0).Databases(0)** returns the first database opened. This concept applies only to Microsoft Access.

current index

For an indexed table-type **Recordset** object, the index most recently set with the **Index** property. This index is the basis for ordering records in a table-type **Recordset**, and is used by the **Seek** method to locate records. A **Recordset** object can have more than one index but can use only one index at a time (although a **TableDef** object may have several indexes defined on it). The Microsoft Jet database engine may use more than one index to evaluate a query.

current record

The record in a **Recordset** object that you can use to modify or examine data. Use the Move methods to reposition the current record in a recordset. Use the Find methods (with a dynaset- or snapshot-type **Recordset** object) or the **Seek** method (with a table-type **Recordset** object) to change the current record position according to specific criteria. Only one record in a **Recordset** can be the current record; however, a **Recordset** may have no current record. For example, after a dynaset-type **Recordset** record has been deleted, or when a **Recordset** has no records, the current record is undefined. In this case, operations that refer to the current record result in a trappable error.

current transaction

All changes made to a **Recordset** object after you use the last **BeginTrans** method and before you use the **Rollback** or **CommitTrans** method.

D

[Back to top](#)

DAO object

An object that is defined by the Data Access Objects (DAO) library. You can use DAO objects, such as the **Database**, **TableDef**, **Recordset**, and **QueryDef** objects, to represent objects that are used to organize and manipulate data, such as tables and queries, in code.

Data Access Objects (DAO)

A programming interface to access and manipulate database objects.

data source

A named Open Database Connectivity (ODBC) resource that specifies the location, driver type, and other parameters needed by an ODBC driver to access an ODBC database. A data source can be any source of database information.

data type

The characteristics of a variable that determine what kind of data the variable can hold. Data types include **Big Integer**, **Binary**, **Byte**, **Boolean**, **Char**, **Currency**, **Date**, **Decimal**, **Double**, **Float**, **Guid**, **Integer**, **Long**, **Long Binary (OLE Object)**, **Memo**, **Numeric**, **Single**, **string**, **Text**, **Time**, **TimeStamp**, **VarBinary**, **Variant** (default), user-defined types (created with the **Type** statement), and object data types, which include host-defined object data types and DAO object types.

database engine

That part of the database system that retrieves data from and stores data in user and system databases.

database replication

The process of reproducing a database so that two or more copies (replicas) of the same database can stay synchronized. Changes to the data in a replicated table in one replica are sent and applied to the other replicas in the replica set. Changes made to the design of the database in the Design Master are sent and applied to all replicas in the set.

data-definition query

An SQL-specific query that can create, alter, or delete a table, or create or delete an index in a database. ANSI defines these as DDL queries and uses the tokens CREATE, DROP, and ALTER.

date literal

Any sequence of characters with a valid format that is surrounded by number signs (#). Valid formats include the date format specified by the locale settings for your code or the universal date format. For example, #12/31/92# is the date literal that represents December 31, 1992 where English-U.S. is the locale setting for your application. Use date literals to maximize portability across national languages.

dates and times

Dates and times are stored internally as different parts of a real number. The value to the left of the decimal represents a date between December 30, 1899 and December 30, 9999, inclusive. Negative values represent dates prior to December 30, 1899. The value to the right of the decimal represents

a time between 0:00:00 and 23:59:59, inclusive. Midday is represented by .5.

DDL (Data Definition Language)

The language used to describe attributes of a database, especially tables, fields, indexes, and storage strategy. ANSI defines this to have the tokens CREATE, DROP, and ALTER.

Decimal data type

A data type that stores a signed, exact numeric value with precision p and scale s ($1 \leq p \leq 15$; $0 \leq s \leq p$).

default synchronization partner

A Synchronizer that is used to automatically synchronize a replica with other replicas in the set. When a replica is created, the Synchronizer managing the source replica becomes the default synchronization partner for the new replica.

default synchronization schedule

A template used by a Synchronizer to define a schedule between a newly managed replica and its default synchronization partner, when the newly managed replica has no existing synchronization schedule. Once the schedule is defined, you can change the schedule.

default workspace

The **Workspace** object that DAO automatically establishes when your application first references any DAO object. This **Workspace** is referenced by **DBEngine.Workspaces(0)** or simply **Workspaces(0)**.

delete query

An action query that deletes a set of rows that match the criteria you specify. A delete query does not return rows.

Design Master

A database to which system tables, system fields, and replication properties have been added. A Design Master is the first replica in a replica set. You can make changes to the database structure only with the Design Master. Replicas in the same replica set can take turns being the Design Master, but there can be only one Design Master at a time in each replica set.

design time

The time during which you build an application in the development environment by adding controls, setting control or form properties, and so on. In contrast, during run time, you interact with the application as a user would.

destination field

A field in a query that will contain the results of your query.

direct synchronization

A type of synchronization that occurs when the two replicas can be opened simultaneously during the synchronization process.

DLL (dynamic-link library)

A set of routines that can be called from procedures and are loaded and linked into your application at run time.

Double data type

A fundamental data type that holds double-precision floating-point numbers in IEEE format. A **Double** variable is stored as a 64-bit (8-byte) number ranging in value from -1.79769313486231E308 to -4.94065645841247E-324 for negative values, from 4.94065645841247E-324 to 1.79769313486231E308 for positive values, and 0.

dropbox folder

The shared network location where a Synchronizer can leave or pick up design and data changes.

dynamic cursor

A dynamic set of rows that you can use to add, change, or delete rows from an underlying database table or tables. A dynamic cursor can contain columns from one or more tables in a database. Membership is not fixed.

dynaset

A type of **Recordset** object that returns a dynamic set of pointers to live database data. Like a table- or a snapshot-type **Recordset**, a dynaset returns data in records (rows) and fields (columns). Unlike a table-type **Recordset**, a dynaset-type **Recordset** can be the result of a query that joins two or more tables. The records in a dynaset-type **Recordset** object are updatable if the **Updatable** property of the **Recordset** is **True**, the **Field** being changed is updatable, and the data page containing the current record is not locked. The data page is locked when the **Update** method is used (when the **LockEdits** property is **False**), or the **Edit** method is used (when the **LockEdits** property is **True**).

E

[Back to top](#)

equi-join

A join in which records from two tables are combined and added to the **Recordset** object only when there are equal values in the joined fields. Also referred to as an inner-join.

exclusive

A type of access that protects data in a database shared over a network. When you open a database in exclusive mode, you prevent others from opening the database.

expression

Any combination of operators, constants, literal values, functions, and names of fields, controls, and properties that evaluates to a single value. You can use expressions as settings for many properties and action arguments, to set criteria, or define calculated fields in queries.

external database

Either an ODBC database such as Microsoft SQL Server that resides on a remote server, or one of the external databases such as Paradox, dBASE, Microsoft FoxPro, Microsoft Excel, Microsoft Access, Lotus 1-2-3, HTML, and Text.

external table

A table in a database outside the open Microsoft Jet database. An external table may take the form of another file or it may be a table in another relational database.

F

[Back to top](#)

field

A category of information stored in a table in a database or a column of data. An element of a database table that contains a specific item of information, such as last name.

field data types

The following are the **Field** data types - Big Integer, Binary, Boolean, Byte, Char, Currency, Date/Time, Decimal, Double, Float, GUID, Integer, Long, Long Binary (OLE Object), Memo, Numeric, Single, Text, Time, TimeStamp, and VarBinary.

field properties

Attributes of a field that determine what kind of data it contains. **Size** and **Type** are examples.

filter

A set of criteria applied to records in order to create a subset of the records.

Float data type

A data type that stores a signed, approximate numeric value with mantissa

precision 15 (zero or absolute value 10-308 to 10308).

foreign key

One or more table fields that refer to the primary key field or fields in another table. A foreign key indicates how the tables are related—the data in the foreign key and primary key fields must match. For example, a list of valid part numbers would contain a foreign key to an inventory table containing references to valid part numbers. Used when establishing referential integrity for a database.

foreign table

A table that provides a foreign key to another table in the database. Generally, you use a foreign table to establish or enforce referential integrity. The foreign table is usually on the "many" side of a one-to-many relationship. An example of a foreign table is a table of customer orders.

form

A window or dialog box. Forms are containers for controls. A multiple-document interface (MDI) form can also act as a container for child forms and some controls.

forward scroll

Move toward the end (**EOF**) of a **Recordset** object.

forward-only cursor

A result set where the current position can only move forward by a specified number of records, or to the very last record. The current position cannot be moved back to the beginning of the result set or to previous records. The cursor membership, order, and values are generally fixed when the cursor is opened. If other users update, delete, or insert rows, the cursor does not reflect these changes until you close and reopen it.

forward-only-type Recordset

A **Recordset** object in which records can be searched only from beginning to end; the current record position cannot be moved back to the first record. Forward-only-type recordsets reduce processing overhead on remote databases. For example, you can use a forward-only-type **Recordset** on a linked remote table to quickly process data in one pass, such as when you are building a customized report.

FTP

File Transfer Protocol. A protocol that makes it possible for a user to transfer files from one location to another over the Internet. URLs of files on FTP servers begin with **ftp://**.

Function procedure

A procedure that performs a specific task within a Microsoft Visual Basic

program and returns a value. A **Function** procedure begins with a **Function** statement and ends with an **End Function** statement.

G

[Back to top](#)

globally unique identifier (GUID)

A 16-byte field used to establish a unique identifier for replication. GUIDs are used extensively to identify replicas, replica sets, tables, records, and other objects.

group

A collection of user accounts in a **Workgroup** object, identified by group name and personal identifier (PID). Permissions assigned to a group apply to all users in the group.

GUID data type

Globally Unique Identifier/Universally Unique Identifier. A unique identification string used with remote procedure calls. Every interface and object class uses a **GUID** for identification. A **GUID** is a 128-bit value. For example, 12345678-1234-1234-1234-123456789ABC is a syntactically correct **GUID**. The **GUIDs** on the client and server must match for the client and server to bind. Vendors of objects can request that Microsoft allocate one or more sets of 256 **GUIDs** for their exclusive use.

Alternatively, if you have a network card, you can run a tool named Uuidgen.exe, which provides a set of 256 **GUIDs** based on the time of day, the date, and a unique number contained in your network card.

H

[Back to top](#)

HTTP

Hypertext Transfer Protocol. The behind-the-scenes Internet protocol that delivers information by way of the World Wide Web. The protocol makes it possible for a user to use a client program to enter a URL (or click a hyperlink) and retrieve text, graphics, sound, and other digital information from a Web server. URLs of files on Web servers begin with **http://**.

hub server

A computer used in a star topology. A replica residing on this computer can be used as a synchronization partner with other replicas.

hyperlink data type

A field data type you use for fields that will store hyperlinks. Hyperlink fields contain text or combinations of text and numbers stored as text that are used as a hyperlink address. A hyperlink address can have up to three parts (each part can contain up to 2048 characters and uses the syntax:*displaytext#address#subaddress*)

I

[Back to top](#)

Identity attribute

Added to allow for syntactical equivalence to SQL server. Only available when using SQL DDL from the Microsoft OLE DB Provider for Jet. This data type has the same functionality as the Counter and AutoNumber data types. It is a synonym and has no additional functionality.

inconsistent

The state of a multiple-table **Recordset** object that enables you to update all fields (columns). For example, in a **Recordset** created by joining two tables in a one-to-many join (as in a Customers and Orders table), you can update **Orders.CustomerID** so that it does not match **Customers.CustomerID**, unless referential integrity disables the update.

increment

Optional parameter used with the **Counter** or **Identity** data types. Available in SQL DDL only when utilizing the Microsoft OLE DB Provider for Jet. This is the value that is used to assign the next value of a new row in a table that contains a **Counter** or **Identity** data type. The increment is the second of the two optional parameters to the **Counter** or **Identity** data types.

index

A dynamic cross-reference of one or more table data fields (columns) that permits faster retrieval of specific records from a table. As records are added, changed, or deleted, the database management system automatically updates the index to reflect the changes. When used with a table-type **Recordset** object, the current index determines the order in which data records are returned to the **Recordset**. A table may have several indexes defined for its data. A DAO **Index** object represents an index for a **Tabledef** object.

indirect synchronization

A type of synchronization that occurs when one Synchronizer leaves changes in a dropbox folder for another Synchronizer to pick up and apply to the replica it manages.

inner join

A join in which records from two tables are combined and added to a **Recordset** only if the values of the joined fields meet a specified condition. For instance, an equi-join is an inner join in which the values of the joined fields must be equal.

installable ISAM

A driver you can specify that allows access to external database formats such as dBASE, Microsoft Excel, and Paradox. ISAM is an acronym for Indexed Sequential Access Method. The Microsoft Jet database engine installs (loads) these ISAM drivers when referenced by your application. The location of these drivers is maintained in the Microsoft Windows Registry.

instance

Any one of a set of objects sharing the same class. For example, multiple instances of a **Form** class share the same code and are loaded with the same controls with which the **Form** class was designed. During run time, the individual properties of controls on each instance can be set to different values.

Integer data type

A fundamental data type that holds integer numbers. An **Integer** variable is stored as a 16-bit (2-byte) number ranging in value from -32,768 to 32,767.

Internet

A worldwide network of thousands of smaller computer networks and millions of commercial, educational, government, and personal computers. The Internet is like an electronic city with virtual libraries, storefronts, business offices, art galleries, and so on.

internet synchronization

A type of synchronization that relies on a series of message exchanges between replicas that are left in a dropbox folder. Only the replica managed by the Synchronizer on the Internet Server has a dropbox. This dropbox is accessible via an HTTP or FTP connection.

intranet

A network within an organization that uses Internet technologies (such as the HTTP or FTP protocols). You can use an intranet to navigate between objects, documents, Web pages, and other destinations using hyperlinks.

intrinsic constant

A constant provided by an application. For example, Data Access Objects (DAO) constants are listed in the DAO object library and can be viewed by using the Object Browser.

J

[Back to top](#)

join

A database operation that combines some or all records from two or more tables, such as an equi-join, inner-join, outer join, or self-join. Generally, a join refers to an association between a field in one table and a field of the same data type in another table. You create a join with an SQL statement. When you define a relationship between two tables, you create a join by specifying the primary and foreign table fields. When you add a table to a query, you need to create a join between appropriate fields in the SQL statement that defines the query.

K

[Back to top](#)

keyset cursor

A set of rows that you can use to add, change, or delete rows from an underlying database table or tables. Movement within the keyset is unrestricted. A keyset cursor can contain columns from one or more tables in a database. Membership is fixed.

L

[Back to top](#)

left join

A left outer join includes all of the records from the first (left) of two tables, even if there are no matching values for records in the second (right) table.

left outer join

An outer join in which all the records from the left side of the LEFT JOIN operation in the query's SQL statement are added to the **Recordset** object, even if there are no matching values in the joined field from the table on the right. Records from the table on the right are combined with those from the table on the left only when there are matching values in the joined fields.

When a left-side record has no match, a row of **Null** values is joined on the right side.

linked table

A table in another database linked to a Microsoft Jet database. Data for linked tables remains in the external database where it can be manipulated by other applications. (Formerly known as *attached tables*.)

local object

A table or query that remains in the replica where it was created. Neither the object nor changes to the object are dispersed to other members in the replica set.

locked

The condition of a data page, **Recordset** object, or **Database** object that makes it read-only to all users except the one who is currently entering data in it.

Long Binary data type

A type of field that can hold 2.14 GB of data. Long Binary fields can contain any type of binary data. Also known as an OLE Object data type in Microsoft Access.

Long data type

A fundamental data type that holds long-integer numbers. A **Long** variable is stored as a 32-bit (4-byte) number ranging in value from -2,147,483,648 to 2,147,483,647.

M

[Back to top](#)

make-table query

An action query that creates a new table from the **Recordset** object of an existing query.

managed replicas

A replica with an assigned Synchronizer. The Synchronizer delivers changes to and receives changes from other replicas in the set.

many-only table

A table that does not appear as the "one" side of any join in a query.

Memo data type

A field data type. Such fields can contain up to 2.14 GB of text data.

message

A packet of information passed from one application to another.

method

A procedure similar to a function that operates on specific objects.

Microsoft Jet database

A database created with the Microsoft Jet database engine. The file name extension for a Microsoft Jet database is .mdb.

Microsoft Jet database engine

A database management system that retrieves data from and stores data in user and system databases. The Microsoft Jet database engine can be thought of as a data manager component with which other data access systems, such as Microsoft Access and Microsoft Visual Basic, are built.

Microsoft Jet workspace

A workspace that uses the Microsoft Jet database engine to access a data source. The data source can be a Microsoft Jet database file (.mdb), an ODBC database, such as a Paradox database, or an ISAM database.

Microsoft Jet-connected ODBC data source

An ODBC data source that is accessed by using Data Access Objects (DAO) and the Microsoft Jet database engine.

multiuser database

A database that permits more than one user to access and modify the same set of data at the same time. In some cases, the additional user may be another instance of your application or another application running on your system that accesses the same data as some other application.

N

[Back to top](#)

normalize

To minimize the duplication of information in a relational database through effective table design.

Null

A value that indicates missing or unknown data. **Null** values can be entered in fields for which information is unknown and in expressions and queries. In Visual Basic, the **Null** keyword indicates a **Null** value. Some fields, such as those defined as containing the primary key, cannot contain **Null** values.

null field

A field containing no characters or values. A null field is not the same as a zero-length string ("") or a field with a value of 0. A field is set to null when the content of the field is unknown. For example, a Date Completed field in

a task table would be left null until a task is completed.

Numeric data type

A data type that stores a signed, exact numeric value with precision p and scale s (1 = p = 15; 0 = s = p).

numeric expression

Any expression that results in a number. This can be any combination of variables, constants, functions, and operators that evaluates to a number.

O

[Back to top](#)

Object Browser

A dialog box you can use to examine the contents of an object library to get information about the objects provided.

object expression

An expression that specifies a particular object. This expression can include any of the object's containers. For example, if your application has an **Application** object that contains a **Document** object that contains a **Text** object, the following are valid object expressions:

Application.Document.Text
Application.Text
Document.Text
Text

object library

A dynamic-link library (DLL) with one or more type library resources that typically has the extension .olb. You can use the Object Browser to examine the contents of an object library to get information about the objects provided.

object replication

The process of reproducing an object (table or query) so that two or more copies of the same object are synchronized. Changes made to the design of the object at the Design Master are sent and applied to all replicas in the set.

object variable

A variable that contains a reference to an object.

ODBC (Open Database Connectivity)

A standard protocol that permits applications to connect to a variety of external database servers or files. ODBC drivers used by the Microsoft Jet database engine permit access to Microsoft SQL Server and several other external databases. The ODBC application programming interface (API) may also be used to access ODBC drivers and the databases they connect to without using the Microsoft Jet database engine.

ODBC data source

A term used to refer to a database or database server used as a source of data. ODBC data sources are referred to by their Data Source Name (DSN). Data sources can be created by using the Microsoft Windows Control Panel or the **RegisterDatabase** method.

ODBC driver

A dynamic-link library (DLL) used to connect a specific Open Database Connectivity data source with another (client) application.

ODBC Driver Manager

An application that manages connections between ODBC-enabled data sources and the drivers used to access them.

OLE

Microsoft's object-based technology that enables you to create applications that contain components from various other applications.

OLE Object data type

A field data type you use for objects created in other applications that can be linked or embedded in a Microsoft Jet database. For example, you could use an OLE Object field to store a collection of pictures.

on-demand synchronization

The exchange of data between two replicas that occurs at the explicit request of the user.

one-to-many relationship

An association between two tables in which the primary key value of each record in the primary table corresponds to the value in the matching field or fields of many records in the related table, or the primary key value of each record in the related table corresponds to the value in the matching field or fields of one and only one record in the primary table.

one-to-one relationship

An association between two tables in which the primary key value of each record in the primary table corresponds to the value in the matching field or fields of one and only one record in the related table, or the primary key value of each record in the related table corresponds to the value in the matching field or fields of one and only one record in the primary table.

optimistic

A type of locking in which the data page containing one or more records, including the record being edited, is unavailable to other users only while the record is being updated by the **Update** method, but is available between the **Edit** and **Update** methods. Optimistic locking is used when accessing ODBC databases or when the **LockEdits** property of the **Recordset** object

is set to **False**.

outer join

A join in which all the records from one table are added to the **Recordset** object, even if there are no matching values in the joined field from the second table. Records from the second table are combined with those from the first table only when there are matching values in the joined fields. Null records are joined when there is no match.

P

[Back to top](#)

page

A portion of the database in which record data is stored. Depending on the size of the records, a page may contain more than one record. In Microsoft? Jet databases (.mdb), a page is 2048 (2K) bytes in length. Jet 4.X file formats have a 4K page (4096 bytes).

parameter

An element containing a value that you can change to affect the results of the query. For example, a query returning data about an employee might have a parameter for the employee's name. You can then use one **QueryDef** object to find data about any employee by setting the parameter to a specific name before running the query. Using the Microsoft? OLE DB Provider for Jet in Microsoft Jet version 4.X, parameters can be exposed from procedures using the ANSI question mark (?) parameter markers. Using SQL syntax, you can create the equivalent to a **QueryDef**.

parameter query

A query that requires you to provide one or more criteria values, such as Redmond for City, before the query is run. A parameter query is not, strictly speaking, a separate kind of query; rather, it extends the flexibility of other queries. Using the Microsoft OLE DB Provider for Jet in Microsoft Jet version 4.X, parameters can be exposed from procedures using the ANSI question mark (?) parameter markers. Using SQL syntax, you can create the equivalent to a **QueryDef**.

parse

To identify the parts of a statement or expression and then validate those parts against the appropriate language rules.

partial replica

A database that contains only a subset of the records in a full replica. With a

partial replica, you can set filters and identify relationships that define which subset of the records in the full replica should be present in the database.

pass-through query

An SQL-specific query you use to send commands directly to a SQL database server (such as Microsoft SQL Server). With pass-through queries, you work with the tables on the server instead of linking them. Pass-through queries are used to execute SQL queries and system-specific commands written by using SQL dialects known only to the server. A pass-through query may or may not return records. If it does, they are always returned in a snapshot.

permission

One or more attributes that specify what kind of access a user has to data or objects in a database. For example, if a user has read data permission for a table or query, the user can view or retrieve but not edit data in the table or query.

persistent object

An object stored in the database; for example, a database table or **QueryDef** object. Dynaset-type or snapshot-type **Recordset** objects are not considered persistent objects because they are created in memory as needed.

personal identifier (PID)

A case-sensitive alphanumeric string 4-20 characters long that the Microsoft Jet database engine uses in combination with the account name to identify a user or group in a **Workgroup** object. You provide the PID and the account name when you create a new user or group.

pessimistic

A type of locking in which the page containing one or more records, including the record being edited, is unavailable to other users when you use the **Edit** method, and remains unavailable until you use the **Update** method. Pessimistic locking is enabled when the **LockEdits** property of the **Recordset** object is set to **True**.

primary key

One or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific records in one table from another table. A primary key is called a foreign key when it is referred to from another table. You can have only one primary key. An Employees table, for example, could use the social security number for the primary key.

primary table

The "one" side of two related tables in a one-to-many relationship with a foreign table. Generally, a primary key table is used to establish or enforce referential integrity.

priority

A long value indicating the priority of the replica for use during conflict resolution. A valid priority value for a Global replica is between 0 and 100, inclusive. Local and Anonymous replicas always have a priority of 0.

During synchronization, if a conflict occurs, the replica with the highest priority will win. The losing data will be logged in the conflict table.

procedural query

An SQL statement that executes a stored procedure.

procedure

Similar to a view except that the SQL syntax can allow for SELECT statements (row returning queries) to pass parameters and allow for DML (non-row returning queries) UPDATE, INSERT, DELETE, and SELECT INTO. Literally the same as a querydef, except that it can be created from SQL syntax without the use of the DAO object model.

property

A named attribute of an object. Properties define object characteristics such as size, color, and screen location, or the state of an object, such as enabled or disabled.

pseudo index

A dynamic cross-reference of one or more table data fields (columns) that permits an ODBC table (server table) without a unique index to be edited. A pseudo index is not an actual index on the server table, but a cross-reference that the Microsoft Jet database engine can use to support table updates. A pseudo index is necessary only if an ODBC table does not have a unique index, which allows the table to be updated. You do not need a pseudo index if the table already has a unique index or if there is no need to update the table. (Tables cannot have a unique index if they were intended to be read-only.) In previous versions, pseudo indexes were called *index specifications*.

Q

[Back to top](#)

query

A formalized instruction to a database to either return a set of records or perform a specified action on a set of records as specified in the query. For example, the following SQL query statement returns records: `SELECT CompanyName FROM Publishers WHERE Region = 'NY'` You can create and run select, action, crosstab, parameter, and SQL-specific queries.

R

[Back to top](#)

read-only

A type of access to data whereby information can be retrieved but not modified.

record

A set of related data about a person, place, event, or other item. Table data is stored in records (rows) in the database. Each record is composed of a set of related fields (columns). Each field defines one attribute of information for the record. Taken together, a record defines one specific unit of retrievable information in a database. A record is also known as a row.

referential integrity

Rules that you set to establish and preserve relationships between tables when you add, change, or delete records. Enforcing referential integrity prohibits users from adding records to a joined table for which there is no primary key, changing values in a primary table that would result in orphaned records in a joined table, and deleting records from a primary table when there are matching related records. If you select the **dbRelationDeleteCascade** or **dbRelationUpdateCascade** option for a relationship, the Microsoft Jet database engine allows changes and deletions, but changes or deletes related records to make sure the rules are still enforced. Microsoft Jet version 4.X, through the Microsoft OLE DB Provider for Jet, exposes a much richer set of ANSI SQL and also exposes the ANSI cascade NULL referential action.

relationship

An association established between common fields (columns) in two tables. A relationship can be one-to-one, many-to-many, or one-to-many.

replica

A copy of a database, including its tables, queries, forms, reports, macros, and modules. A replica is a member of a replica set and can be synchronized with other replicas in the set. Changes to the data in a

replicated table in one replica are sent and applied to the other members in the replica set.

replica set

Replicas that share the same database design and unique replica set identifier. Synchronization occurs between replica set members.

replica set member

A replica that belongs to a replica set. Members of a replica set share the same database design and can be synchronized with one another.

replica set topology

The order in which changes are propagated from replica to replica. Topology is important because it determines how quickly changes in another replica appear in your replica.

replicate

Produce a replica, or copy, of something— for example, a database.

replicated database

A database to which additional tables, fields, and properties have been added to record information about changes to data and the design of replicated objects in the database.

replicated object

A table or query that is in all replicas in the replica set. You can change the replicated object only at the Design Master. These changes are then dispersed to other replicas in the replica set during a synchronization.

replication

The process of creating and synchronizing replicas in separate locations. Replication adds system fields, system tables, and replication properties to a database so that changes made to the data in or to the design of a database are sent and applied to all the members of the replica set.

requery

To rerun a query to reflect changes to the records, retrieve newly added records, and eliminate deleted records.

reserved word

A word that is part of a language, such as the Microsoft Jet database engine SQL language. Reserved words include the names of statements, predefined functions and data types, methods, operators, and objects. Examples include SELECT, UPDATE, BETWEEN, SET, and INSERT. Do not use reserved words for variable or object names. When using the Microsoft OLE DB Provider for Jet, the reserved word list for Microsoft Jet version 4.X has been increased to be compliant with ANSI SQL-92.

retention period

The number of days the system will retain certain replication system information for a replica. If the retention period expires before a replica is synchronized, the synchronization will fail.

right join

A right outer join includes all of the records from the second (right) of two tables, even if there are no matching values for records in the first (left) table. For example, you could use LEFT JOIN with the Departments (left) and Employees (right) tables to select all departments, including those that have no employees assigned to them. To select all employees, including those who are not assigned to a department, you would use RIGHT JOIN.

right outer join

An outer join in which all the records from the right side of the RIGHT JOIN operation in the query's SQL statement are added to the **Recordset** object, even if there are no matching values in the joined field from the left table. Records from the left table are combined with those from the right table only when there are matching values in the joined fields.

rollback

When using the Microsoft OLE DB Provider for Jet, you can now start (BeginTrans), commit (CommitTrans), or rollback (RollBack) a transaction without going through the object model DAO or ADO. With rollback (RollBack), a transaction that was started (BeginTrans) can be ended or removed.

row

A set of related columns or fields used to hold data. A row is synonymous with a record in the Microsoft Jet database engine. A table is composed of zero or more rows of data.

row-level locking

A new feature in Microsoft Jet version 4.X that allows a user to lock a row instead of a page when updating or deleting a row. Row-level locking is not implemented on columns that are indexed or LV (long value) data types (Memo).

row-level tracking

Row-level tracking is a mechanism where the entire row participates in conflict determination and resolution. A conflict will occur if more than one replica modifies the same row, even if the changed data exists in different columns of that row.

run time

The time when code is running. During run time, you interact with the code as a user would.

run-time error

An error that occurs when code is running. A run-time error results when a statement attempts an invalid operation.

S

[Back to top](#)

satellite computer

In a star topology, a computer on which a replica resides. This replica is copied from the base computer and synchronizes with a replica on the hub server by default.

schedule

The times when the Synchronizer will initiate synchronization with other replicas. The minimum increment of time between scheduled synchronizations is 15 minutes. You can set different schedules between any two Synchronizers.

scheduled synchronization

An exchange of data between two replicas that occurs at a predetermined time.

scope

The attribute of a variable or procedure that determines which sections of which modules recognize it. There are three levels of scope: public, module, and procedure. Variables that you declare with **Public** can be accessed by any module, while variables that you declare in a specific module can be used only within that module. Also, variables that you declare in a **Sub** or **Function** procedure can be used only in that particular procedure.

security-enabled workgroup

A Microsoft Jet database **Workgroup** object to which users log on with a user name and password and in which access to database objects is restricted according to permissions specified for user accounts and groups.

security

A set of features used to specify or restrict the access that specified users or user groups have to data and objects in a database.

seed

Optional parameter used with the **Counter** or **Identity** data types. Available in SQL DDL only when utilizing the Microsoft OLE DB Provider for Jet. Can also be referred to as the starting value. The seed is the first of

the two optional parameters to the **Counter** or **Identity** data types. The following SQL statement is an example with the parameters: CREATE TABLE Customers (CustId IDENTITY (100, 10) CONSTRAINT pkCustomers PRIMARY KEY, CFrstNm VARCHAR(10), CLstNm VARCHAR(15)); The column CustId in the first row will have a value of 100. The second row will have a value of 110. When the optional parameters are utilized and do not contain values of (1,1), Jet will not reset the seed to the maximum value in the table. This behavior is different than in previous versions of Jet. If the optional parameters are not passed or a value of (1,1) is not used, when the database is compacted the seed will be reset to the maximum value of the **Counter** or **Identity** column in the table.

select query

A query that asks a question about the data stored in your tables and returns a **Recordset** object without changing the data. Once the **Recordset** data is retrieved, you can examine and make changes to the data in the underlying tables. In contrast, action queries can make changes to your data, but they do not return data records.

self-join

A join in which records from a table are combined with other records from the same table when there are matching values in the joined fields. A self-join can be an inner join or an outer join. A self-join is useful when you ask hierarchical questions, such as a query that shows all the managers and the people who report to them.

server

The database management system designed to share data with client applications; servers and clients are often connected over a network. A database server usually contains and manages a central repository of data that remote client applications can retrieve and manipulate.

server-side cursor

Cursors that reside on the server, as opposed to residing on the client computer. While client-side cursors copy the cursor to the workstation, server-side cursors use the resources of the database server to maintain the cursors.

session

A session delineates a sequence of operations performed by the Microsoft Jet database engine. A session begins when a user logs on and ends when a user logs off. All operations performed during a session form one transaction scope and are subject to permissions determined by the logon user name and password. Sessions are implemented as **Workspace** objects

by DAO.

shared network folder

The shared network location where a Synchronizer can leave or pick up design and data changes.

Short data type

A Microsoft Jet database engine fundamental data type that corresponds to the **Integer** data type defined by Microsoft Visual Basic for Applications. A **Short** variable is stored as a 16-bit (2-byte) number ranging in value from -32,768 to 32,767.

Single data type

A fundamental data type that holds single-precision floating-point numbers in IEEE format. A **Single** variable is stored as a 32-bit (4-byte) number ranging in value from -3.402823E38 to -1.401298E-45 for negative values, from 1.401298E-45 to 3.402823E38 for positive values, and 0.

snapshot

A fixed copy of a set of records retrieved from the database and copied into memory. A snapshot derived from a Microsoft Jet-connected data source cannot be updated. Snapshot-type **Recordset** objects can be created from a base table, a query, or another **Recordset**. All SQL pass-through queries return snapshots.

sort order

A sequencing principle used to order data, alphabetically or numerically. The sort order can be either ascending or descending.

SQL statement/string

An expression that defines a Structured Query Language (SQL) command, such as SELECT, UPDATE, or DELETE, and may include clauses such as WHERE and ORDER BY. SQL strings and statements are typically used in queries, **Recordset** objects, and aggregate functions but can also be used to create or modify a database structure.

SQL-specific query

A query that can be created only by writing an SQL statement. Subqueries and pass-through, union, and data-definition queries are SQL-specific queries.

standard deviation

A parameter that indicates the way in which a probability function is centered around its mean and that is equal to the square root of the moment in which the deviation from the mean is squared. Used to describe sets of values to determine how far the value varies from the arithmetic mean or average.

standard naming conventions

A set of specific rules for naming DAO objects. Names can be up to 64 characters long and can include any combination of letters, numbers, spaces, and special characters except a period (.), an exclamation point (!), a backquote character (`), and brackets ([]). You also cannot use leading spaces or control characters (ASCII values 0 through 31).

star topology

A type of organizational map for networked computers. For replication, it refers to a hub server, base computer, and the distribution of members of the replica set on other satellite computers. Topology is important for determining synchronization schedules.

static cursor

A result set where the membership, order, and values are generally fixed when the cursor is opened. If other users update, delete, or insert rows, the cursor does not reflect these changes until you close and reopen it.

string comparison

The use of an operator to determine whether one string is greater than or equal to another string. If you use **Option Compare Text** in the Declarations section of a module, string comparisons are not case-sensitive. If you use **Option Compare Binary**, comparisons are case-sensitive. If you use **Option Compare Database**, the comparison method is set by the current database.

String data type

A fundamental data type that holds character information. A **String** variable is either fixed-length or variable-length and contains one character per byte. Fixed-length strings are declared to be a specific length and can contain 1 to approximately 64K (2^{16}) characters. Variable-length strings can be any length up to 2 billion (2^{31}) characters (approximately 64K [2^{16}] characters for Microsoft Windows version 3.1 and earlier), less a small amount of storage overhead.

string expression

Any expression that evaluates to a sequence of contiguous characters. Elements of the expression can include a function that returns a string, a string literal, a string constant, a string variable, a string **Variant**, or a function that returns a string **Variant (VarType 8)**.

Structured Query Language (SQL)

A language used in querying, updating, and managing relational databases. SQL can be used to retrieve, sort, and filter specific data to be extracted from the database. You can use SQL SELECT statements anywhere a table

name, query name, or field name is accepted. For example, you can use an SQL statement in place of a table name in the **OpenRecordset** method.

subquery

A subquery is a SELECT statement nested inside a SELECT, SELECT...INTO, INSERT...INTO, DELETE, or UPDATE statement or inside another subquery.

synchronization

The process of updating two replicas in which all updated records and objects are exchanged. The exchange of data between two replicas can be one-way or two-way and may be handled by a Synchronizer.

synchronization conflict

A situation in which users of two replicas have changed the same record. One of the updated records is selected as the "winner," and the record in the "losing" replica is updated to contain this data. The other updated record is placed in a conflict table in the losing replica.

synchronization error

A problem that occurs during synchronization that prevents changes to a record from being completed successfully. This is more serious than a conflict and should be resolved as soon as possible.

syntax error

An error that occurs when you enter a line of code that Microsoft Visual Basic does not recognize.

system object

A database object defined by the Microsoft Jet database engine, such as the table MSysIndexes.

T

[Back to top](#)

table

The basic unit of data storage in a relational database. A table stores data in records (rows) and fields (columns) and is usually about a particular category of things, such as employees or parts. Also called a base table. A table-type **Recordset** is a logical representation of a base table in a database.

TEMP

A TEMP environment variable is set by your autoexec.bat file when you start your system. Generally, TEMP points to an area on your hard disk

used by Microsoft Windows and other programs, like the Microsoft Jet database engine, to store information that does not need to be saved after you shut down your system. For example, the following line in your autoexec.bat file points the TEMP environment variable to the D:\TempArea folder: SET TEMP=D:\TempArea

temporary disk

The disk or directory, identified by the TEMP operating system environment variable, where the Microsoft Jet database engine stores temporary files and result sets. Also known as a temporary drive. Although the TEMP environment variable can point to a RAM disk, this is not recommended.

Text(n) data type (with length specification)

A field data type. Text fields can contain up to 255 characters or the number of characters specified by the **Size** property of the **Field** object, whichever is less. If the **Size** property of the text field is set to 0, the text field can hold up to 255 characters of data.

Text data type (without length specification)

A field data type. Text fields defined without an explicit length are the same as a Memo field and can contain up to 2.14 GB of character data.

Time data type

A data type that stores a time value. The value is dependent on the clock setting of the data source.

transaction

A series of changes made to a database's data and schema. Mark the beginning of a transaction with the **BeginTrans** statement, commit the transaction by using the **CommitTrans** statement, and undo all your changes since **BeginTrans** by using the **Rollback** statement. Transactions are optional and can be nested up to five levels. Transactions increase the speed of operations that change data and enable you to reverse changes easily. Transactions are global to the referenced database object's **Workspace**.

U

[Back to top](#)

Unicode representation format

A 16-bit character set managed by the Unicode Consortium that contains all of the characters needed to display every language in the world. Unicode is

increasingly being used to facilitate multi-lingual operation of software.

union query

An SQL-specific select query that creates a snapshot-type **Recordset** object containing data from all specified records in two or more tables with any duplicate records removed. To include the duplicates, add the keyword **ALL**. For instance, a union query of the Customers table and the Suppliers table results in a snapshot-type **Recordset** that contains all suppliers that are also customers.

unmanaged replica

A replica that is not managed by a Synchronizer. Unmanaged replicas appear on the Synchronization window with a special icon. You cannot set synchronization schedules from an unmanaged replica, although they can be synchronized by another Synchronizer or through DAO code.

update

The process that saves changes to data in a record. Until the record is saved, changes are stored in a temporary record called the copy buffer. The **UPDATE** clause in an SQL statement changes data values in one or more records (rows) in a database table.

update query

An action query that changes a set of records according to criteria you specify. An update query does not return any records.

user account

An account identified by a user name and personal identifier (PID) that is created to manage access to database objects in a Microsoft Jet database **Workgroup** object.

V

[Back to top](#)

validation

The process of checking whether entered data meets certain conditions or limitations.

validation properties

Properties used to set conditions on table fields and records. Validation properties include **ValidationRule**, **Required** and **AllowZeroLength**.

validation rule

A rule that sets limits or conditions on what can be entered in one or more fields. Validation rules can be set for a **Field** or **TableDef** object. Validation

rules are checked when you update a record containing fields requiring validation. If the rule is violated, a trappable error results.

VarBinary data type

A data type that stores variable-length binary data. The maximum length is 255 bytes.

variable

A named storage location that can contain data that can be modified during program execution. Each variable has a name that uniquely identifies it within its level of scope. A data type can be specified or not.

variance

The square of the standard deviation. It is a measure of the amount by which all values in a group vary from the average value of the group.

Variant data type

A special data type that can contain numeric, string, or date data as well as the special values **Empty** and **Null**. The **VarType** function defines how the data in a **Variant** is treated. All variables become variant types if not explicitly declared as some other type.

variant expression

Any expression that can evaluate to numeric, string, or date data as well as the special values **Empty** and **Null**

view

View is the ANSI definition of a virtual table. A view is synonymous to a stored query in Access that is comprised of a SELECT statement without parameters. A view is a stored definition that is materialized at execution. It is different from a table in that it does not physically store any data. It returns data only. A view can be created from the new ANSI SQL CREATE VIEW syntax exposed through the Microsoft OLE DB Provider for Jet. Alternatively, a procedure can be created to allow for SQL DML statements and SELECT statements that have parameters. This can be done with the new CREATE PROCEDURE syntax exposed through the Microsoft OLE DB Provider for Jet.

W

[Back to top](#)

wildcard characters

The asterisk (*), percent sign (%), question mark (?), underscore character (_), number sign (#), exclamation point (!), hyphen (-), and brackets ([])

are wildcard characters. You can use these characters in queries and expressions to include all records, file names, or other items that begin with specific characters or match a certain pattern. You can also use wildcard characters and matching characters to further refine a search when using an SQL statement.

workgroup

A group of users in a multiuser environment who share data and the same workgroup information file (typically the System.mdw file).

workgroup administrator

A member of the group Admins in a Microsoft Jet database **Workgroup** object. A workgroup administrator can always obtain full permissions for any database object created in the **Workgroup**.

Z

[Back to top](#)

zero-length string

A string containing no characters (""). The **Len** function of a zero-length string returns 0.

What's New in Microsoft Jet 4.0

Microsoft® Jet version 4.x has many new features, including additional SQL reserved words, SQL data types, and ODBC scalar functions. There is now more information comparing SQL with ANSI SQL and more information about equivalent ANSI SQL data types. ANSI SQL wildcards, the percent sign (%) and the underscore character (_), are available with Jet 4.x and the Microsoft OLE DB Provider for Jet.

For detailed information about these new features, see:

[SQL Reserved Words](#) [Comparison of Microsoft Jet SQL and ANSI SQL](#)
[SQL Data Types](#) [Equivalent ANSI SQL Data Types](#)
[ODBC Scalar Functions](#) [Using Wildcard Characters in String Comparisons](#)

Microsoft Jet 4.x has new data definition language statements, new data manipulation language statements, additional methods for customizing Microsoft Windows® Registry settings, and additional settings for initializing the database engine driver.

For detailed information about these new features, see:

[What's New with Data Definition Language](#) [What's New with Windows Registry Settings](#)
[What's New with Data Manipulation Language](#)

What's New with Jet 4.0 Data Definition Language

Microsoft® Jet version 4.x has new and enhanced statements for the data definition language.

For detailed information about these statements, see:

[ALTER TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CONSTRAINT Clause](#)

[ADD USER Statement](#)

[CREATE PROCEDURE Statement](#)

[DROP USER or GROUP Statement](#)

[CREATE TABLE Statement](#)

[ALTER USER or DATABASE Statement](#)

[CREATE VIEW Statement](#)

[GRANT Statement](#)

[DROP Statement](#)

[REVOKE Statement](#)

See Also

[Overview of What's New](#)

[What's New with Windows Registry Settings](#)

[What's New with Data Manipulation Language](#)

What's New with Jet 4.0 Data Manipulation Language

Microsoft® Jet version 4.x has new and enhanced statements for the data manipulation language.

For detailed information about these statements, see:

[EXECUTE Statement](#) [TRANSACTION Statement](#)
[PROCEDURE Clause](#)

See Also

[Overview of What's New](#)

[What's New with Windows Registry Settings](#)

[What's New with Data Definition Language](#)

What's New with Jet 4.0 Windows Registry Settings

Microsoft® Jet version 4.x now offers four methods to modify your Microsoft Windows® Registry settings. In addition to using Regedit.exe to overwrite default settings or create a portion in your application's registry tree to manage the settings, you can also use the **SetOption** method from DAO or the connection properties in the Microsoft OLE DB Provider for Jet.

Detailed information and examples are provided for initialization settings for the Jet database engine. These settings are used for access to Microsoft Access databases. Information is also provided for Microsoft Jet replication settings.

For detailed information about these methods and settings, see:

[Customizing Windows Registry Settings for Microsoft Jet](#)

[Initializing the Microsoft Jet 4.0 Database Engine Driver](#)

See Also

[Overview of What's New](#)

[What's New with Data Manipulation Language](#)

[What's New with Data Definition Language](#)

About Microsoft Jet 4.0 Service Packs

Microsoft Jet 4.0 Service Packs starting with Service Pack 8 include an updated sandbox mode that allows Microsoft Office Access 2003 to block potentially unsafe expressions. Some features in Office Access 2003 will not function properly if you do not install Microsoft Jet 4.0 Service Pack 8 or later.

Features that will not function properly without Microsoft Jet 4.0 Service Pack 8 or later

The following features will not function properly if Microsoft Jet 4.0 Service Pack 8 or later is not installed:

- Wizards may not start.
- Switchboards created by **Switchboard Manager** won't function properly.
- You can't use the **RunCode** macro action to call a Visual Basic for Applications (VBA) procedure.
- Event handlers that call VBA procedures won't function.
- VBA procedures that are called from a property sheet or used in a SQL statement won't function.
- The following VBA functions won't work when called from a property sheet or used in a SQL statement:
 - FormatCurrency
 - FormatDateTime
 - FormatNumber
 - FormatPercent
 - InStrB
 - InStrRev
 - MonthName
 - Replace
 - StrReverse
 - WeekDayName

SQL Aggregate Functions

Using the [SQL aggregate functions](#) , you can determine various statistics on sets of values. You can use these functions in a query and aggregate expressions in the SQL property of a QueryDef object or when creating a Recordset object based on an SQL query.

[Avg Function](#)

[Count Function](#)

[First, Last Functions](#)

[Min, Max Functions](#)

[StDev, StDevP Functions](#)

[Sum Function](#)

[Var, VarP Functions](#)

SQL Expressions

An [SQL expression](#) is a [string](#) that makes up all or part of an [SQL statement](#) . For example, the FindFirst method on a Recordset object uses an SQL expression consisting of the selection criteria found in an SQL WHERE clause .

The [Microsoft Jet database engine](#) uses the Microsoft® Visual Basic® for Applications (or VBA) expression service to perform simple arithmetic and function evaluation. All of the operators used in Microsoft Jet SQL expressions (except [Between](#) , [In](#) , and [Like](#)) are defined by the VBA expression service. In addition, the VBA expression service offers over 100 VBA functions that you can use in SQL expressions. For example, you can use these VBA functions to compose an SQL query in the Microsoft Access query Design view, and you can also use these functions in an SQL query in the [DAO](#) OpenRecordset method in Microsoft Visual C++®, Microsoft Visual Basic, and Microsoft Excel code.

SQL Reserved Words

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

The following list includes all words reserved by the [Microsoft Jet database engine](#) for use in [SQL statements](#) . The words in the list that are not in all uppercase letters are also reserved by other applications. Consequently, the individual Help topics for these words provide general descriptions that do not focus on [SQL](#) usage.

Note Words followed by an asterisk (*) are reserved but currently have no meaning in the context of a Microsoft® Jet SQL statement (for example, **Level** and **TableID**). Words that are not underlined do not have linked explanations.

A

ABSOLUTE	ANY
ADD	ARE
ADMINDB	AS
ALL	ASC
Alphanumeric — See TEXT	ASSERTION
ALTER	AUTHORIZATION
ALTER TABLE	AUTOINCREMENT — See COUNTER
And	Avg
AS	

B-C

BEGIN	COLLATION
Between	COLUMN
BINARY	COMMIT
BIT	COMP, COMPRESSION
BIT_LENGTH	CONNECT
BOOLEAN — See BIT	CONNECTION

BOTH	CONSTRAINT, CONSTRAINTS
BY	CONTAINER
BYTE	CONTAINS
CASCADE	CONVERT
CATALOG	Count
CHAR, CHARACTER — See TEXT COUNTER	
CHAR_LENGTH	CREATE
CHARACTER_LENGTH	CURRENCY
CHECK	CURRENT_DATE
CLOSE	CURRENT_TIME
CLUSTERED	CURRENT_TIMESTAMP
COALESCE	CURRENT_USER
COLLATE	CURSOR

D

DATABASE	DISALLOW
DATE — See DATETIME	DISCONNECT
DATETIME	DISTINCT
DAY	DISTINCTROW
DEC, DECIMAL	DOMAIN
DECLARE	DOUBLE
DELETE	DROP
DESC	

E-H

Eqv	FOREIGN
EXCLUSIVECONNECT	FROM
EXEC, EXECUTE	FROM Clause
EXISTS	GENERAL — See LONGBINARY
EXTRACT	GRANT

FALSE
FETCH
FIRST
FLOAT, FLOAT8 — See DOUBLE
FLOAT4 — See SINGLE

GROUP
GUID
HAVING

I

IDENTITY
IEEEDOUBLE — See DOUBLE
IEEESINGLE — See SINGLE
IGNORE
IMAGE
Imp
In
IN
INDEX
INDEXCREATEDB
INNER
INPUT
INSENSITIVE
INSERT
INSERT INTO
INT, INTEGER, INTEGER4 — See LONG
INTEGER1 — See BYTE
INTEGER2 — See SHORT
INTERVAL
INTO
Is
ISOLATION

J-M

JOIN
KEY
LANGUAGE
LAST
LEFT
Level*
Like
LOGICAL, LOGICAL1 — See BIT
LONG
LONGBINARY
LONGTEXT
LOWER
MATCH
Max
MEMO — See LONGTEXT
Min
MINUTE
Mod
MONEY — See CURRENCY
MONTH

LONGCHAR

N-P

NATIONAL	Outer*
NCHAR	OUTPUT
NONCLUSTERED	OWNERACCESS
Not	PAD
NTEXT	PARAMETERS
NULL	PARTIAL
NUMBER — See DOUBLE	PASSWORD
NUMERIC — See DECIMAL	PERCENT
NVARCHAR	PIVOT
OCTET_LENGTH	POSITION
OLEOBJECT — See LONGBINARY	PRECISION
ON	PREPARE
OPEN	PRIMARY
OPTION	PRIVILEGES
Or	PROC, PROCEDURE
ORDER	PUBLIC

Q-S

REAL — See SINGLE	SMALLDATETIME
REFERENCES	SMALLINT — See SHORT
RESTRICT	SMALLMONEY
REVOKE	SOME
RIGHT	SPACE
ROLLBACK	SQL
SCHEMA	SQLCODE, SQLERROR, SQLSTATE
SECOND	StDev
SELECT	StDevP

SELECTSCHEMA	STRING — See TEXT
SELECTSECURITY	SUBSTRING
SET	Sum
SHORT	SYSNAME
SINGLE	SYSTEM_USER
SIZE	

T-Z

TABLE	UPDATEOWNER
TableID*	UPDATESECURITY
TEMPORARY	UPPER
TEXT	USAGE
TIME — See DATETIME	USER
TIMESTAMP	USING
TIMEZONE_HOUR	VALUE
TIMEZONE_MINUTE	VALUES
TINYINT	Var
TO	VARBINARY — See BINARY
TOP	VARCHAR — See TEXT
TRAILING	VarP
TRANSACTION	VARYING
TRANSFORM	VIEW
TRANSLATE	WHEN
TRANSLATION	WHENEVER
TRIM	WHERE
TRUE	WITH
UNION	WORK
UNIQUE	Xor
UNIQUEIDENTIFIER	YEAR
UNKNOWN	YESNO — See BIT
UPDATE	ZONE
UPDATEIDENTITY	

SQL Data Types

The [Microsoft Jet database engine SQL](#) data types consist of 13 primary data types defined by the Microsoft® Jet database engine and several valid synonyms recognized for these data types.

The following table lists the primary data types.

Data type	Storage size	Description
BINARY	1 byte per character	Any type of data may be stored in a field of this type. No translation of the data (for example, to text) is made. How the data is input in a binary field dictates how it will appear as output.
BIT	1 byte	Yes and No values and fields that contain only one of two values.
TINYINT	1 byte	An integer value between 0 and 255.
MONEY	8 bytes	A scaled integer between – 922,337,203,685,477.5808 and 922,337,203,685,477.5807.
DATETIME (See DOUBLE)	8 bytes	A date or time value between the years 100 and 9999.
UNIQUEIDENTIFIER	128 bits	A unique identification number used with remote procedure calls.
REAL	4 bytes	A single-precision floating-point value with a range of – 3.402823E38 to – 1.401298E-45 for negative values, 1.401298E-45 to 3.402823E38 for positive values, and 0.
FLOAT	8 bytes	A double-precision floating-point value with a range of – 1.79769313486232E308 to – 4.94065645841247E-324 for negative values, 4.94065645841247E-324 to 1.79769313486232E308 for positive values, and 0.

SMALLINT	2 bytes	A short integer between – 32,768 and 32,767. (See Notes)
INTEGER	4 bytes	A long integer between – 2,147,483,648 and 2,147,483,647. (See Notes)
DECIMAL	17 bytes	An exact numeric data type that holds values from 10 ²⁸ - 1 through - 10 ²⁸ - 1. You can define both precision (1 - 28) and scale (0 - defined precision). The default precision and scale are 18 and 0, respectively.
TEXT	2 bytes per character	Zero to a maximum of 2.14 gigabytes. (See Notes)
IMAGE	As required	Zero to a maximum of 2.14 gigabytes. Used for OLE objects.
CHARACTER	2 bytes per character	Zero to 255 characters. (See Notes)

Notes

- Both the [seed](#) and the [increment](#) can be modified using an ALTER TABLE statement . New rows inserted into the table will have values, based on the new seed and increment values, that are automatically generated for the column. If the new seed and increment can yield values that match values generated based on the preceding seed and increment, duplicates will be generated. If the column is a [primary key](#) , then inserting new rows may result in errors when duplicate values are generated.
- To find the last value that was used for an auto-increment column, you can use the following statement: SELECT @@IDENTITY. You cannot specify a table name. The value returned is from the last table, containing an auto-increment column, that was updated.
- Characters in fields defined as either [TEXT](#) (also known as MEMO) or [CHAR](#) (also known as TEXT(n) with a specific length) are stored in [the](#)

[Unicode representation format](#) . Unicode characters uniformly require two bytes to store each character. For existing Microsoft Jet databases that contain predominately character data, this could mean that the database file would nearly double in size when converted to the Microsoft Jet 4.0 format. Yet Unicode representation of many character sets, those formerly denoted as SBCS (Single-Byte Character Sets) can easily be compressed to a single byte. If you define a CHAR column with the COMPRESSION attribute, data will automatically be compressed as it is stored and uncompressed when it is retrieved from the column.

See Also

[Equivalent ANSI SQL Data Types](#)

SQL Subqueries

A subquery is a SELECT statement nested inside a SELECT, SELECT...INTO, INSERT...INTO, DELETE, or UPDATE statement or inside another subquery.

Syntax

You can use three forms of syntax to create a subquery:

comparison [ANY | ALL | SOME] (*sqlstatement*)

expression [NOT] IN (*sqlstatement*)

[NOT] EXISTS (*sqlstatement*)

A subquery has these parts:

Part	Description
<i>comparison</i>	An expression and a comparison operator that compares the expression with the results of the subquery.
<i>expression</i>	An expression for which the result set of the subquery is searched.
<i>sqlstatement</i>	A SELECT statement, following the same format and rules as any other SELECT statement. It must be enclosed in parentheses.

Remarks

You can use a subquery instead of an expression in the field list of a SELECT statement or in a WHERE or HAVING clause. In a subquery, you use a SELECT statement to provide a set of one or more specific values to evaluate in the WHERE or HAVING clause expression.

Use the ANY or SOME predicate, which are synonymous, to retrieve records in the main query that satisfy the comparison with any records retrieved in the subquery. The following example returns all products whose unit price is greater than that of any product sold at a discount of 25 percent or more:

```
SELECT * FROM Products
```

```
WHERE UnitPrice > ANY
(SELECT UnitPrice FROM OrderDetails
WHERE Discount >= .25);
```

Use the ALL predicate to retrieve only those records in the main query that satisfy the comparison with all records retrieved in the subquery. If you changed ANY to ALL in the previous example, the query would return only those products whose unit price is greater than that of all products sold at a discount of 25 percent or more. This is much more restrictive.

Use the IN predicate to retrieve only those records in the main query for which some record in the subquery contains an equal value. The following example returns all products with a discount of 25 percent or more:

```
SELECT * FROM Products
WHERE ProductID IN
(SELECT ProductID FROM OrderDetails
WHERE Discount >= .25);
```

Conversely, you can use NOT IN to retrieve only those records in the main query for which no record in the subquery contains an equal value.

Use the EXISTS predicate (with the optional NOT reserved word) in true/false comparisons to determine whether the subquery returns any records.

You can also use table name [aliases](#) in a subquery to refer to tables listed in a FROM clause outside the subquery. The following example returns the names of employees whose salaries are equal to or greater than the average salary of all employees having the same job title. The Employees table is given the alias "T1":

```
SELECT LastName,
FirstName, Title, Salary
FROM Employees AS T1
WHERE Salary >=
```

```
(SELECT Avg(Salary)
FROM Employees
WHERE T1.Title = Employees.Title) Order by Title;
```

In the preceding example, the AS [reserved word](#) is optional.

Some subqueries are allowed in [crosstab queries](#) — specifically, as predicates (those in the WHERE clause). Subqueries as output (those in the SELECT list) are not allowed in crosstab queries.

See Also

ALL, DISTINCT, DISTINCTROW, TOP Predicates	SELECT Statement
DELETE Statement	SELECT INTO Statement
HAVING Clause	UNION Operation
INNER JOIN Operation	UPDATE Statement
INSERT INTO Statement	WHERE Clause
LEFT JOIN, RIGHT JOIN Operations	

ODBC Scalar Functions

Microsoft® Jet SQL supports the use of the [ODBC](#) defined syntax for scalar functions. For example, the query:

```
SELECT DAILYCLOSE, DAILYCHANGE FROM DAILYQUOTE  
WHERE {fn ABS(DAILYCHANGE)} > 5
```

Would return all rows where the absolute value of the change in the price of a stock was greater than five.

A subset of the ODBC defined scalar functions is supported. The following table lists the functions that are supported.

For a description of the arguments and a complete explanation of the escape syntax for including functions in a SQL statement, see the ODBC documentation.

String Functions

ASCII LENGTH RTRIM
CHAR LOCATE SPACE
CONCAT LTRIM SUBSTRING
LCASE RIGHT UCASE
LEFT

Numeric Functions

ABS FLOOR SIN
ATAN LOG SQRT
CEILING POWER TAN
COS RAND MOD
EXP SIGN

Time & Date Functions

CURDATE	DAYOFYEAR	MONTH
CURTIME	YEAR	WEEK
NOW	HOUR	QUARTER
DAYOFMONTH	MINUTE	MONTHNAME
DAYOFWEEK	SECOND	DAYNAME

Data Type Conversion

String literals can be converted to the following data types:

CONVERT	SQL_FLOAT, SQL_DOUBLE, SQL_NUMERIC, SQL_INTEGER, SQL_REAL, SQL_SMALLINT, SQL_VARCHAR and SQL_DATETIME.
---------	--

See Also

[Configuring the Microsoft Jet Database Engine for ODBC Access](#)

Calculating Fields in SQL Functions

You can use the [string expression](#) argument in an SQL aggregate function to perform a calculation on values in a field. For example, you could calculate a percentage (such as a surcharge or sales tax) by multiplying a field value by a fraction.

The following table provides examples of calculations on fields from the Orders and Order Details tables in the Northwind.mdb database.

Calculation	Example
Add a number to a field	Freight + 5
Subtract a number from a field	Freight - 5
Multiply a field by a number	UnitPrice * 2
Divide a field by a number	Freight / 2
Add one field to another	UnitsInStock + UnitsOnOrder
Subtract one field from another	ReorderLevel - UnitsInStock

The following example calculates the average discount amount of all orders in the Northwind.mdb database. It multiplies the values in the UnitPrice and Discount fields to determine the discount amount of each order and then calculates the average. You can use this expression in an [SQL statement](#) in Visual Basic code:

```
SELECT Avg(UnitPrice * Discount) AS [Average Discount] FROM [Order Details];
```

See Also

[SQL Aggregate Functions](#) [SQL Expressions](#)

Comparison of Microsoft Jet SQL and ANSI SQL

[Microsoft Jet database engine SQL](#) is generally [ANSI -89 Level 1](#) compliant. However, certain ANSI SQL features are not implemented in Microsoft® Jet SQL. With the release of Microsoft Jet version 4.X, the Microsoft OLE DB Provider for Jet exposes more [ANSI-92 SQL](#) syntax. Conversely, Microsoft Jet SQL includes reserved words and features not supported in ANSI SQL.

Major Differences

- Microsoft Jet SQL and ANSI SQL each have different reserved words and data types. For more information, see [Microsoft Jet Database Engine SQL Reserved Words](#) and [Equivalent ANSI SQL Data Types](#) . Using the Microsoft OLE DB Provider for Jet with Jet 4.X, there are additional reserved words.
- Different rules apply to the [Between...And](#) construct, which has the following syntax:

expr1 [NOT] **Between** *value1* **And** *value2*

In Microsoft Jet SQL, *value1* can be greater than *value2*; in ANSI SQL, *value1* must be equal to or less than *value2*.

- Microsoft Jet SQL supports both ANSI SQL wildcard characters and Microsoft Jet-specific [wildcard characters](#) to use with the **Like** operator. The use of the ANSI and Microsoft Jet wildcard characters is mutually exclusive. You must use one set or the other and cannot mix them. The ANSI SQL wildcards are only available when using Jet 4.X and the Microsoft OLE DB Provider for Jet. If you try to use the ANSI SQL wildcards through Microsoft Access or DAO, then they will be interpreted as literals. The opposite is true when using the Microsoft OLE DB Provider for Jet and Jet 4.X.

Matching character	Microsoft Jet SQL	ANSI SQL
Any single character	?	_ (underscore)

Zero or more characters *

%

- Microsoft Jet SQL is generally less restrictive. For example, it permits grouping and ordering on expressions.
- Microsoft Jet SQL supports more powerful expressions.

Enhanced Features of Microsoft Jet SQL

Microsoft Jet SQL provides the following enhanced features:

The [TRANSFORM](#) statement, which provides support for [crosstab queries](#) .

- Additional SQL Aggregate Functions , such as **StDev** and **VarP**.

The [PARAMETERS](#) declaration for defining [parameter queries](#) .

ANSI SQL Features Not Supported in Microsoft Jet SQL

Microsoft Jet SQL does not support the following ANSI SQL features:

- DISTINCT aggregate function references. For example, Microsoft Jet SQL does not allow SUM(DISTINCT *columnname*).
- The LIMIT TO *nn* ROWS clause used to limit the number of rows returned by a query. You can use only the [WHERE Clause](#) to limit the scope of a query.

See Also

[Equivalent ANSI SQL Data Types](#)

Like

[Microsoft Jet Database Engine SQL Data Types](#)

[Microsoft Jet Database Engine SQL Reserved Words](#)

[SQL Aggregate Functions](#)

[Using Wildcard Characters in String Comparisons](#)

Equivalent ANSI SQL Data Types

The following table lists ANSI [SQL](#) data types, their equivalent [Microsoft Jet database engine](#) SQL data types, and their valid synonyms. It also lists the equivalent Microsoft® SQL Server™ data types.

ANSI SQL data type	Microsoft Jet SQL data type	Synonym	Microsoft SQL Server data type
BIT, BIT VARYING	BINARY (See Notes)	VARBINARY, BINARY VARYING BIT VARYING	BINARY, VARBINARY
Not supported	BIT (See Notes)	BOOLEAN, LOGICAL, LOGICAL1, YESNO	BIT
Not supported	TINYINT	INTEGER1, BYTE	TINYINT
Not supported	COUNTER (See Notes)	AUTOINCREMENT (See Notes)	
Not supported	MONEY	CURRENCY	MONEY
DATE, TIME, TIMESTAMP	DATETIME	DATE, TIME (See Notes)	DATETIME
Not supported	UNIQUEIDENTIFIER	GUID	UNIQUEIDENTIFIER
DECIMAL	DECIMAL	NUMERIC, DEC	DECIMAL
REAL	REAL	SINGLE, FLOAT4, IEEESINGLE	REAL
DOUBLE PRECISION, FLOAT	FLOAT	DOUBLE, FLOAT8, IEEEDOUBLE, NUMBER (See Notes)	FLOAT
SMALLINT	SMALLINT	SHORT, INTEGER2	SMALLINT
INTEGER	INTEGER	LONG, INT, INTEGER4	INTEGER
INTERVAL	Not supported		Not supported
		LONGBINARY,	

Not supported	IMAGE	GENERAL, OLEOBJECT	IMAGE
Not supported	TEXT (See Notes)	LONGTEXT, LONGCHAR, MEMO, NOTE, NTEXT (See Notes) TEXT(n), ALPHANUMERIC, CHARACTER, STRING, VARCHAR, CHARACTER VARYING, NCHAR, NATIONAL CHARACTER, NATIONAL CHARACTER VARYING	TEXT
CHARACTER, CHARACTER VARYING, NATIONAL CHARACTER, NATIONAL CHARACTER VARYING	CHAR (See Notes)	NATIONAL CHARACTER, NATIONAL CHAR, NATIONAL CHARACTER VARYING, NATIONAL CHAR VARYING (See Notes)	CHAR, VARCHAR, NCHAR, NVARCHAR

Notes

- The ANSI SQL BIT data type does not correspond to the Microsoft Jet SQL BIT data type. It corresponds to the BINARY data type instead. There is no ANSI SQL equivalent for the Microsoft Jet SQL BIT data type.
- TIMESTAMP is no longer supported as a synonym for DATETIME.
- NUMERIC is no longer supported as a synonym for FLOAT or DOUBLE. NUMERIC is now used as a synonym for DECIMAL.
- A LONGTEXT field is always stored in the [Unicode representation format](#).
- If the data type name [TEXT](#) is used without specifying the optional length, for example TEXT(25), a LONGTEXT field is created. This enables CREATE TABLE statements to be written that will yield data types consistent with Microsoft SQL Server.

- A CHAR field is always stored in the [Unicode representation format](#) , which is the equivalent of the ANSI SQL NATIONAL CHAR data type.
- If the data type name [TEXT](#) is used and the optional length is specified, for example TEXT(25), the data type of the field is equivalent to the CHAR data type. This preserves backwards compatibility for most Microsoft Jet applications, while enabling the [TEXT data type](#) (without a length specification) to be aligned with Microsoft SQL Server.

See Also

[Microsoft Jet Database Engine SQL Data Types](#)

Using Wildcard Characters in String Comparisons

Built-in pattern matching provides a versatile tool for making string comparisons. The following table shows the wildcard characters you can use with the **Like** operator and the number of digits or strings they match.

Character(s) in <i>pattern</i>	Matches in <i>expression</i>
? or _ (underscore)	Any single character
* or %	Zero or more characters
#	Any single digit (0— 9)
[<i>charlist</i>]	Any single character in <i>charlist</i>
[! <i>charlist</i>]	Any single character not in <i>charlist</i>

You can use a group of one or more characters (*charlist*) enclosed in brackets ([]) to match any single character in *expression*, and *charlist* can include almost any characters in the [ANSI](#) character set, including digits. You can use the special characters opening bracket ([), question mark (?), number sign (#), and [asterisk](#) (*) to match themselves directly only if enclosed in brackets. You cannot use the closing bracket (]) within a group to match itself, but you can use it outside a group as an individual character.

In addition to a simple list of characters enclosed in brackets, *charlist* can specify a range of characters by using a hyphen (-) to separate the upper and lower bounds of the range. For example, using [A-Z] in *pattern* results in a match if the corresponding character position in *expression* contains any of the uppercase letters in the range A through Z. You can include multiple ranges within the brackets without delimiting the ranges. For example, [a-zA-Z0-9] matches any alphanumeric character.

It is important to note that the ANSI SQL wildcards (%) and (_) are only available with Microsoft® Jet version 4.X and the Microsoft OLE DB Provider for Jet. They will be treated as literals if used through Microsoft Access or DAO.

Other important rules for pattern matching include the following:

- An exclamation mark (!) at the beginning of *charlist* means that a match is made if any character except those in *charlist* are found in *expression*. When used outside brackets, the exclamation mark matches itself.
- You can use the hyphen (-) either at the beginning (after an exclamation mark if one is used) or at the end of *charlist* to match itself. In any other location, the hyphen identifies a range of ANSI characters.
- When you specify a range of characters, the characters must appear in ascending sort order (A-Z or 0-100). [A-Z] is a valid pattern, but [Z-A] is not.
- The character sequence [] is ignored; it is considered to be a [zero-length string](#) (“”).

See Also

[Like Operator SQL Expressions](#)

CREATE TABLE Statement

Creates a new table.

Note The [Microsoft Jet database engine](#) does not support the use of CREATE TABLE, or any of the [DDL](#) statements, with non-Microsoft Jet database engine databases. Use the DAO Create methods instead.

Syntax

```
CREATE [TEMPORARY] TABLE table (field1 type [(size)] [NOT NULL]
[WITH COMPRESSION | WITH COMP] [index1] [, field2 type [(size)] [NOT
NULL] [index2] [, ...]] [, CONSTRAINT multifieldindex [, ...]])
```

The CREATE TABLE statement has these parts:

Part	Description
<i>table</i>	The name of the table to be created.
<i>field1, field2</i>	The name of field or fields to be created in the new table. You must create at least one field.
<i>type</i>	The data type of <i>field</i> in the new table.
<i>size</i>	The field size in characters (Text and Binary fields only).
<i>index1, index2</i>	A CONSTRAINT clause defining a single-field index.
<i>multifieldindex</i>	A CONSTRAINT clause defining a multiple-field index.

Remarks

Use the CREATE TABLE statement to define a new table and its fields and field constraints. If NOT NULL is specified for a field, then new records are required to have valid data in that field.

A CONSTRAINT clause establishes various restrictions on a field, and can be used to establish the [primary key](#) . You can also use the CREATE INDEX statement to create a primary key or additional indexes on existing tables.

You can use NOT NULL on a single field or within a named CONSTRAINT

clause that applies to either a single field or to a multiple-field named CONSTRAINT. However, you can apply the NOT NULL restriction only once to a field. Attempting to apply this restriction more than once results in a run-time error.

When a TEMPORARY table is created it is visible only within the session in which it was created. It is automatically deleted when the session is terminated. Temporary tables can be accessed by more than one user.

The WITH COMPRESSION attribute can be used only with the CHARACTER and MEMO (also known as TEXT) data types and their synonyms.

The WITH COMPRESSION attribute was added for CHARACTER columns because of the change to the Unicode character representation format. Unicode characters uniformly require two bytes for each character. For existing Microsoft® Jet databases that contain predominately character data, this could mean that the database file would nearly double in size when converted to the Microsoft Jet version 4.0 format. However, [Unicode representation](#) of many character sets, those formerly denoted as Single-Byte Character Sets (SBCS) can easily be compressed to a single byte. If you define a CHARACTER column with this attribute, data will automatically be compressed as it is stored and uncompressed when retrieved from the column.

MEMO columns can also be defined to store data in a compressed format. However, there is a limitation. Only instances of MEMO columns that, when compressed, will fit within 4096 bytes or less, will be compressed. All other instances of MEMO columns will remain uncompressed. This means that within a given table, for a given MEMO column, some data may be compressed and some data may not be compressed.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[CREATE INDEX statement](#)

[CREATE PROCEDURE Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

CREATE INDEX Statement

Creates a new index on an existing table.

Note For non-Microsoft Jet databases, the [Microsoft Jet database engine](#) does not support the use of CREATE INDEX (except to create a [pseudo index](#) on an [ODBC linked table](#)) or any of the [data definition language \(DDL\)](#) statements. Use the [DAO Create methods](#) instead. For more information see the Remarks section.

Syntax

```
CREATE [ UNIQUE ] INDEX index
ON table (field [ASC|DESC][, field [ASC|DESC], ...])
[WITH { PRIMARY | DISALLOW NULL | IGNORE NULL }]
```

The CREATE INDEX statement has these parts:

Part	Description
<i>index</i>	The name of the index to be created.
<i>table</i>	The name of the existing table that will contain the index.
<i>field</i>	The name of the field or fields to be indexed. To create a single-field index, list the field name in parentheses following the table name. To create a multiple-field index, list the name of each field to be included in the index. To create descending indexes, use the DESC reserved word; otherwise, indexes are assumed to be ascending.

Remarks

To prohibit duplicate values in the indexed field or fields of different records, use the UNIQUE reserved word.

In the optional WITH clause you can enforce data validation rules. You can:

- Prohibit [Null](#) entries in the indexed field or fields of new records by using

the DISALLOW NULL option.

- Prevent records with **Null** values in the indexed field or fields from being included in the index by using the IGNORE NULL option.
- Designate the indexed field or fields as the [primary key](#) by using the PRIMARY reserved word. This implies that the key is unique, so you can omit the UNIQUE reserved word.

You can use CREATE INDEX to create a [pseudo index](#) on a [linked table](#) in an [ODBC data source](#), such as Microsoft® SQL Server™, that does not already have an index. You do not need permission or access to the remote server to create a pseudo index, and the remote database is unaware of and unaffected by the pseudo index. You use the same syntax for both linked and native tables. Creating a pseudo-index on a table that would ordinarily be read-only can be especially useful.

You can also use the ALTER TABLE statement to add a single- or multiple-field index to a table, and you can use the ALTER TABLE statement or the DROP statement to remove an index created with ALTER TABLE or CREATE INDEX.

Note Do not use the PRIMARY [reserved word](#) when you create a new index on a table that already has a primary key; if you do, an error occurs.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE PROCEDURE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

CREATE PROCEDURE Statement

Creates a stored [procedure](#) .

Note The [Microsoft Jet database engine](#) does not support the use of CREATE PROCEDURE, or any of the [DDL](#) statements, with non-Microsoft Jet database engine databases.

Syntax

```
CREATE PROCEDURE procedure  
[param1 datatype[, param2 datatype[, ...]] AS sqlstatement
```

The CREATE PROCEDURE statement has these parts:

Part	Description
<i>procedure</i>	A name for the procedure. It must follow standard naming conventions . From one to 255 field names or parameters . For example:
<i>param1</i> , <i>param2</i>	CREATE PROCEDURE Sales_By_Country [Beginning Date] DateTime, [Ending Date] DateTime;
<i>datatype</i>	One of the primary Microsoft Jet SQL Data Types or their synonyms.
<i>sqlstatement</i>	An SQL statement such as SELECT, UPDATE, DELETE, INSERT, CREATE TABLE, DROP TABLE, and so on.

Remarks

An SQL procedure consists of a PROCEDURE clause that specifies the name of the procedure, an optional list of parameter definitions, and a single [SQL statement](#) .

A procedure name cannot be the same as the name of an existing table.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX Statement](#)

[CREATE TABLE Statement](#)

[PARAMETERS Declaration](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

CREATE USER or GROUP Statement

Creates one or more new users or groups.

Syntax

Create a user:

```
CREATE USER user password pid [, user password pid, ...]
```

Create a group:

```
CREATE GROUP group pid [, group pid, ...]
```

The CREATE USER or GROUP statement has these parts:

Part	Description
<i>user</i>	The name of a user to be added to the workgroup information file.
<i>group</i>	The name of a group to be added to the workgroup information file.
<i>password</i>	The password to be associated with the specified <i>user</i> name.
<i>pid</i>	The personal id.

Remarks

A *user* and a *group* cannot have the same name.

A *password* is required for each *user* or *group* that is created.

See Also

[ADD USER Statement](#)

[CREATE TABLE Statement](#)

[ALTER USER or DATABASE Statement](#)

[CREATE VIEW Statement](#)

[ALTER TABLE Statement](#)

[DROP Statement](#)

[CONSTRAINT Clause](#)

[DROP USER or GROUP Statement](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

CREATE VIEW Statement

Creates a new [view](#) .

Note The [Microsoft Jet database engine](#) does not support the use of CREATE VIEW, or any of the [DDL](#) statements, with non-Microsoft Jet database engine databases.

Syntax

```
CREATE VIEW view [(field1[, field2[, ...]])] AS selectstatement
```

The CREATE VIEW statement has these parts:

Part	Description
<i>view</i>	The name of the view to be created.
<i>field1, field2</i>	The name of field or fields for the corresponding fields specified in <i>selectstatement</i> .
<i>selectstatement</i>	A SQL SELECT statement.

Remarks

The SELECT statement that defines the view cannot be a SELECT...INTO statement.

The SELECT statement that defines the view cannot contain any parameters.

The name of the [view](#) cannot be the same as the name of an existing table.

If the query defined by the SELECT statement is updatable, then the view is also updatable. Otherwise, the view is read-only.

If any two fields in the query defined by the SELECT statement have the same name, the view definition must include a field list specifying unique names for each of the fields in the query.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[SELECT statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

[SELECT...INTO](#)

ADD USER Statement

Adds one or more existing *users* to an existing *group*.

Syntax

ADD USER *user*[, *user*, ...] TO *group*

The ADD USER statement has these parts:

Part	Description
<i>user</i>	The name of a user to be added to the workgroup information file.
<i>group</i>	The name of a group to be added to the workgroup information file.

Remarks

Once a *user* had been added to a *group*, the *user* has all the permissions that have been granted to the *group*.

See Also

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX statement](#)

[CREATE PROCEDURE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

DROP USER or GROUP Statement

Deletes one or more existing *users* or *groups*, or removes one or more existing *users* from an existing *group*.

Syntax

Delete one or more *users* or remove one or more *users* from a *group*:

```
DROP USER user[, user, ...] [FROM group]
```

Delete one or more *groups*:

```
DROP GROUP group[, group, ...]
```

The DROP USER or GROUP statement has these parts:

Part	Description
<i>user</i>	The name of a user to be removed from the workgroup information file.
<i>group</i>	The name of a group to be removed from the workgroup information file.

Remarks

If the FROM keyword is used in the DROP USER statement, then each of the *users* listed in the statement will be removed from the *group* specified following the FROM keyword. However, the *users* themselves will not be deleted.

The DROP GROUP statement will delete the specified *group(s)*. The *users* who are members of the *group(s)* will not be affected, but they will no longer be members of the deleted *group(s)*.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

ALTER TABLE Statement

Modifies the design of a table after it has been created with the CREATE TABLE statement.

Note The [Microsoft Jet database engine](#) does not support the use of ALTER TABLE, or any of the [data definition language \(DDL\)](#) statements, with non-Microsoft Jet databases. Use the [DAO](#) Create methods instead.

Syntax

```
ALTER TABLE table {ADD {COLUMN field type[(size)] [NOT NULL]
[CONSTRAINT index] |
ALTER COLUMN field type[(size)] |
CONSTRAINT multifieldindex} |
DROP {COLUMN field | CONSTRAINT indexname} }
```

The ALTER TABLE statement has these parts:

Part	Description
<i>table</i>	The name of the table to be altered.
<i>field</i>	The name of the field to be added to or deleted from <i>table</i> . Or, the name of the field to be altered in <i>table</i> .
<i>type</i>	The data type of <i>field</i> .
<i>size</i>	The field size in characters (Text and Binary fields only).
<i>index</i>	The index for <i>field</i> .
<i>multifieldindex</i>	The definition of a multiple-field index to be added to <i>table</i> .
<i>indexname</i>	The name of the multiple-field index to be removed.

Remarks

Using the ALTER TABLE statement you can alter an existing table in several ways. You can:

- Use `ADD COLUMN` to add a new field to the table. You specify the field name, data type, and (for Text and Binary fields) an optional size. For example, the following statement adds a 25-character Text field called Notes to the Employees table:

```
ALTER TABLE Employees ADD COLUMN Notes TEXT(25)
```

You can also define an index on that field.

If you specify `NOT NULL` for a field then new records are required to have valid data in that field.

- Use `ALTER COLUMN` to change the data type of an existing field. You specify the field name, the new data type, and an optional size for Text and Binary fields. For example, the following statement changes the data type of a field in the Employees table called ZipCode (originally defined as Integer) to a 10-character Text field:

```
ALTER TABLE Employees ALTER COLUMN ZipCode TEXT(10)
```

- Use `ADD CONSTRAINT` to add a multiple-field index.
- Use `DROP COLUMN` to delete a field. You specify only the name of the field.
- Use `DROP CONSTRAINT` to delete a multiple-field index. You specify only the index name following the `CONSTRAINT` reserved word.

Notes

- You cannot add or delete more than one field or index at a time.
- You can use the `CREATE INDEX` statement to add a single- or multiple-field index to a table, and you can use `ALTER TABLE` or the `DROP` statement to delete an index created with `ALTER TABLE` or `CREATE INDEX`.
- You can use `NOT NULL` on a single field or within a named `CONSTRAINT` clause that applies to either a single field or to a multiple-field named `CONSTRAINT`. However, you can apply the `NOT NULL` restriction only once to a field. Attempting to apply this restriction more than once results in a run-time error.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX statement](#)

[CREATE PROCEDURE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

ALTER USER or DATABASE Statement

Changes the password for an existing user or for a database.

Syntax

ALTER DATABASE PASSWORD *newpassword oldpassword*

ALTER USER *user* PASSWORD *newpassword oldpassword*

The ALTER USER or DATABASE statement has these parts:

Part	Description
<i>user</i>	The name of a user to be added to the workgroup information file.
<i>newpassword</i>	The new password to be associated with the specified <i>user</i> or <i>database</i> name.
<i>oldpassword</i>	The existing password to be associated with the specified <i>user</i> or <i>group</i> name.

See Also

[ADD USER Statement](#)

[CREATE USER or GROUP Statement](#)

[ALTER TABLE Statement](#)

[CREATE VIEW Statement](#)

[CONSTRAINT Clause](#)

[DROP Statement](#)

[CREATE INDEX Statement](#)

[DROP USER or GROUP Statement](#)

[CREATE PROCEDURE Statement](#)

[GRANT Statement](#)

[CREATE TABLE Statement](#)

[REVOKE Statement](#)

DROP Statement

Deletes an existing table, procedure, or view from a database, or deletes an existing index from a table.

Note The [Microsoft Jet database engine](#)> does not support the use of DROP, or any of the [DDL](#) statements, with non-Microsoft Jet databases. Use the [DAO Delete](#) method instead.

Syntax

```
DROP {TABLE table | INDEX index ON table | PROCEDURE procedure |  
VIEW view}
```

The DROP statement has these parts:

Part	Description
<i>table</i>	The name of the table to be deleted or the table from which an index is to be deleted.
<i>procedure</i>	The name of the procedure to be deleted.
<i>view</i>	The name of the view to be deleted.
<i>index</i>	The name of the index to be deleted from <i>table</i> .

Remarks

You must close the table before you can delete it or remove an index from it.

You can also use ALTER TABLE to delete an index from a table.

You can use CREATE TABLE to create a table and CREATE INDEX or ALTER TABLE to create an index. To modify a table, use ALTER TABLE.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP](#)

Statement

ALTER TABLE Statement

CONSTRAINT Clause

CREATE INDEX Statement

CREATE PROCEDURE Statement

Statement

CREATE VIEW Statement

DROP USER or GROUP Statement

GRANT Statement

REVOKE Statement

GRANT Statement

Grants specific privileges to an existing user or group.

Syntax

```
GRANT {privilege[, privilege, ...]} ON {TABLE table |  
OBJECT object|
```

```
CONTAINER container } TO {authorizationname[, authorizationname, ...]}
```

The GRANT statement has these parts:

Part	Description
	The privilege or privileges to be granted. Privileges are specified using the following keywords:
<i>privilege</i>	SELECT, DELETE, INSERT, UPDATE, DROP, SELECTSECURITY, UPDATESECURITY, DBPASSWORD, UPDATEIDENTITY, CREATE, SELECTSCHEMA, SCHEMA and UPDATEOWNER.
<i>tablename</i>	Any valid table name.
<i>object</i>	This can encompass any non-table object. A stored query (view or procedure) is one example.
<i>container</i>	The name of a valid container.
<i>authorizationname</i>	A user or group name.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[REVOKE Statement](#)

REVOKE Statement

Revokes specific privileges from an existing user or group.

Syntax

```
REVOKE {privilege[, privilege, ...]} ON  
{TABLE table |  
OBJECT object |  
CONTAINER container }  
FROM {authorizationname[, authorizationname, ...]}
```

The REVOKE statement has these parts:

Part	Description
<i>privilege</i>	The privilege or privileges to be revoked. Privileges are specified using the following keywords: SELECT, DELETE, INSERT, UPDATE, DROP, SELECTSECURITY, UPDATESECURITY, DBPASSWORD, UPDATEIDENTITY, CREATE, SELECTSCHEMA, SCHEMA and UPDATEOWNER.
<i>table</i>	Any valid table name.
<i>object</i>	This can encompass any non-table object. A stored query (view or procedure) is one example.
<i>container</i>	The name of a valid container.
<i>authorizationname</i>	A user or group name.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[ALTER TABLE Statement](#)

[CONSTRAINT Clause](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

SELECT Statement

Instructs the [Microsoft Jet database engine](#) to return information from the database as a set of records.

Syntax

```
SELECT [predicate] { * | table.* | [table.]field1 [AS alias1] [, [table.]field2 [AS  
alias2] [, ...]]}  
FROM tableexpression [, ...] [IN externaldatabase]  
[WHERE... ]  
[GROUP BY... ]  
[HAVING... ]  
[ORDER BY... ]  
[WITH OWNERACCESS OPTION]
```

The SELECT statement has these parts:

Part	Description
<i>predicate</i>	One of the following predicates: ALL, DISTINCT, DISTINCTROW, or TOP. You use the predicate to restrict the number of records returned. If none is specified, the default is ALL.
*	Specifies that all fields from the specified table or tables are selected.
<i>table</i>	The name of the table containing the fields from which records are selected.
<i>field1, field2</i>	The names of the fields containing the data you want to retrieve. If you include more than one field, they are retrieved in the order listed.
<i>alias1, alias2</i>	The names to use as column headers instead of the original column names in <i>table</i> .
<i>tableexpression</i>	The name of the table or tables containing the data you want to retrieve. The name of the database containing the tables in

externaldatabase tableexpression if they are not in the current database.

Remarks

To perform this operation, the Microsoft® Jet database engine searches the specified table or tables, extracts the chosen columns, selects rows that meet the criterion, and sorts or groups the resulting rows into the order specified.

SELECT statements do not change data in the database.

SELECT is usually the first word in an [SQL statement](#) . Most SQL statements are either SELECT or SELECT...INTO statements.

The minimum syntax for a SELECT statement is:

SELECT *fields* FROM *table*

You can use an [asterisk](#) (*) to select all fields in a table. The following example selects all of the fields in the Employees table:

```
SELECT * FROM Employees;
```

If a field name is included in more than one table in the FROM clause, precede it with the table name and the . (dot) operator. In the following example, the Department field is in both the Employees table and the Supervisors table. The SQL statement selects departments from the Employees table and supervisor names from the Supervisors table:

```
SELECT Employees.Department, Supervisors.SupvName  
FROM Employees INNER JOIN Supervisors  
WHERE Employees.Department = Supervisors.Department;
```

When a Recordset object is created, the Microsoft Jet database engine uses the table's field name as the Field object name in the **Recordset** object. If you want a different field name or a name is not implied by the expression used to generate the field, use the AS [reserved word](#) . The following example uses the title Birth to name the returned **Field** object in the resulting **Recordset** object:

```
SELECT BirthDate  
  
AS Birth FROM Employees;
```

Whenever you use aggregate functions or queries that return ambiguous or duplicate **Field** object names, you must use the AS clause to provide an alternate name for the **Field** object. The following example uses the title HeadCount to name the returned **Field** object in the resulting **Recordset** object:

```
SELECT COUNT(EmployeeID)  
  
AS HeadCount FROM Employees;
```

You can use the other clauses in a SELECT statement to further restrict and organize your returned data. For more information, see the Help topic for the clause you are using.

See Also

ALL DISTINCT, DISTINCTROW, TOP Predicates	ORDER BY Clause (Microsoft Jet SQL)
DELETE Statement	SELECT...INTO Statement
FROM Clause	SQL Aggregate Functions
GROUP BY Clause	UNION Operation
HAVING Clause	UPDATE Statement
IN Clause	WHERE Clause
INSERT INTO Statement	WITH OWNERACCESS OPTION Declaration

SELECT...INTO Statement

Creates a [make-table query](#) .

Syntax

```
SELECT field1[, field2[, ...]] INTO newtable [IN externaldatabase]  
FROM source
```

The SELECT...INTO statement has these parts:

Part	Description
<i>field1, field2</i>	The name of the fields to be copied into the new table.
<i>newtable</i>	The name of the table to be created. It must conform to standard naming conventions . If <i>newtable</i> is the same as the name of an existing table, a trappable error occurs.
<i>externaldatabase</i>	The path to an external database . For a description of the path, see the IN clause.
<i>source</i>	The name of the existing table from which records are selected. This can be single or multiple tables or a query.

Remarks

You can use make-table queries to archive records, make backup copies of your tables, or make copies to export to another database or to use as a basis for reports that display data for a particular time period. For example, you could produce a Monthly Sales by Region report by running the same make-table query each month.

Notes

- You may want to define a [primary key](#) for the new table. When you create the table, the fields in the new table inherit the [data type](#) and field size of each field in the query's underlying tables, but no other field or table properties are transferred.

- To add data to an existing table, use the INSERT INTO statement instead to create an [append query](#) .
- To find out which records will be selected before you run the make-table query, first examine the results of a SELECT statement that uses the same selection criteria.

See Also

[ALL, DISTINCT, DISTINCTROW, TOP Predicates](#) [SELECT Statement](#)

[FROM Clause](#)

[UNION Operation](#)

[IN Clause](#)

[WHERE Clause](#)

[INSERT INTO Statement](#)

INSERT INTO Statement

Adds a record or multiple records to a table. This is referred to as an [append query](#).

Syntax

Multiple-record append query:

```
INSERT INTO target [(field1[, field2[, ...]])] [IN externaldatabase]  
SELECT [source.]field1[, field2[, ...]]  
FROM tableexpression
```

Single-record append query:

```
INSERT INTO target [(field1[, field2[, ...]])]  
VALUES (value1[, value2[, ...]])
```

The INSERT INTO statement has these parts:

Part	Description
<i>target</i>	The name of the table or query to append records to.
<i>field1, field2</i>	Names of the fields to append data to, if following a <i>target</i> argument, or the names of fields to obtain data from, if following a <i>source</i> argument.
<i>externaldatabase</i>	The path to an external database . For a description of the path, see the IN clause.
<i>source</i>	The name of the table or query to copy records from.
<i>tableexpression</i>	The name of the table or tables from which records are inserted. This argument can be a single table name or a compound resulting from an INNER JOIN, LEFT JOIN, or RIGHT JOIN operation or a saved query.
<i>value1, value2</i>	The values to insert into the specific fields of the new record. Each value is inserted into the field that corresponds to the value's position in the list: <i>value1</i> is inserted into <i>field1</i> of the new record, <i>value2</i> into <i>field2</i> , and so on. You must separate

values with a comma, and enclose text fields in quotation marks (' ').

Remarks

You can use the INSERT INTO statement to add a single record to a table using the single-record append query syntax as shown above. In this case, your code specifies the name and value for each field of the record. You must specify each of the fields of the record that a value is to be assigned to and a value for that field. When you do not specify each field, the default value or **Null** is inserted for missing columns. Records are added to the end of the table.

You can also use INSERT INTO to append a set of records from another table or query by using the SELECT ... FROM clause as shown above in the multiple-record append query syntax. In this case, the SELECT clause specifies the fields to append to the specified *target* table.

The *source* or *target* table may specify a table or a query. If a query is specified, the [Microsoft Jet database engine](#) appends records to any and all tables specified by the query.

INSERT INTO is optional but when included, precedes the SELECT statement.

If your destination table contains a [primary key](#) , make sure you append unique, non-**Null** values to the primary key field or fields; if you do not, the [Microsoft Jet database engine](#) will not append the records.

If you append records to a table with an [AutoNumber](#) field and you want to renumber the appended records, do not include the AutoNumber field in your query. Do include the AutoNumber field in the query if you want to retain the original values from the field.

Use the IN clause to append records to a table in another database.

To create a new table, use the SELECT... INTO statement instead to create a [make-table query](#) .

To find out which records will be appended before you run the append query, first execute and view the results of a [select query](#) that uses the same selection

criteria.

An append query copies records from one or more tables to another. The tables that contain the records you append are not affected by the append query.

Instead of appending existing records from another table, you can specify the value for each field in a single new record using the VALUES clause. If you omit the field list, the VALUES clause must include a value for every field in the table; otherwise, the INSERT operation will fail. Use an additional INSERT INTO statement with a VALUES clause for each additional record you want to create.

See Also

[FROM Clause](#)

[IN Clause](#)

[INNER JOIN Operation](#)

[LEFT JOIN, RIGHT JOIN Operations](#)

[SELECT Statement](#)

[SELECT...INTO Statement](#)

[WHERE Clause](#)

UPDATE Statement

Creates an [update query](#) that changes values in fields in a specified table based on specified criteria.

Syntax

```
UPDATE table  
SET newvalue  
WHERE criteria;
```

The UPDATE statement has these parts:

Part	Description
<i>table</i>	The name of the table containing the data you want to modify.
<i>newvalue</i>	An expression that determines the value to be inserted into a particular field in the updated records.
<i>criteria</i>	An expression that determines which records will be updated. Only records that satisfy the expression are updated.

Remarks

UPDATE is especially useful when you want to change many records or when the records that you want to change are in multiple tables.

You can change several fields at the same time. The following example increases the Order Amount values by 10 percent and the Freight values by 3 percent for shippers in the United Kingdom:

```
UPDATE Orders  
  
SET OrderAmount = OrderAmount * 1.1,  
Freight = Freight * 1.03  
  
WHERE ShipCountry = 'UK';
```

Important

- UPDATE does not generate a result set. Also, after you update records using an update query, you cannot undo the operation. If you want to know which records were updated, first examine the results of a [select query](#) that uses the same criteria, and then run the update query.
- Maintain backup copies of your data at all times. If you update the wrong records, you can retrieve them from your backup copies.

See Also

[SELECT Statement WHERE Clause](#)

DELETE Statement

Creates a [delete query](#) that removes records from one or more of the tables listed in the FROM clause that satisfy the WHERE clause.

Syntax

```
DELETE [table.*]  
FROM table  
WHERE criteria
```

The DELETE statement has these parts:

Part	Description
<i>table</i>	The optional name of the table from which records are deleted.
<i>table</i>	The name of the table from which records are deleted.
<i>criteria</i>	An expression that determines which records to delete.

Remarks

DELETE is especially useful when you want to delete many records.

To drop an entire table from the database, you can use the Execute method with a DROP statement. If you delete the table, however, the structure is lost. In contrast, when you use DELETE, only the data is deleted; the table structure and all of the table properties, such as field attributes and indexes, remain intact.

You can use DELETE to remove records from tables that are in a [one-to-many relationship](#) with other tables. [Cascade delete](#) operations cause the records in tables that are on the many side of the relationship to be deleted when the corresponding record in the one side of the relationship is deleted in the query. For example, in the relationship between the Customers and Orders tables, the Customers table is on the one side and the Orders table is on the many side of the relationship. Deleting a record from Customers results in the corresponding Orders records being deleted if the cascade delete option is specified.

A delete query deletes entire records, not just data in specific fields. If you want to delete values in a specific field, create an [update query](#) that changes the values to **Null** .

Important

- After you remove records using a delete query, you cannot undo the operation. If you want to know which records were deleted, first examine the results of a [select query](#) that uses the same criteria, and then run the delete query.
- Maintain backup copies of your data at all times. If you delete the wrong records, you can retrieve them from your backup copies.

See Also

[DROP Statement](#)

[SELECT Statement](#)

[FROM Clause](#)

[UPDATE Statement](#)

[IN Clause](#)

[WHERE Clause](#)

[INNER JOIN Operation](#)

EXECUTE Statement

Used to invoke the execution of a [procedure](#) .

Syntax

```
EXECUTE procedure [param1[, param2[, ...]]]
```

The EXECUTE statement has these parts:

Part	Description
<i>procedure</i>	The name of the procedure that is to be executed.
<i>param1, param2, ...</i>	Values for the defined by the procedure.

See Also

[PARAMETERS Declaration PROCEDURE Clause](#)

TRANSACTION Statement

Used to initiate and conclude explicit [transactions](#) .

Syntax

Initiate a new transaction.

```
BEGIN TRANSACTION
```

Conclude a transaction by committing all work performed during the transaction.

```
COMMIT [TRANSACTION | WORK]
```

Conclude a transaction by [rolling back](#) all work performed during the transaction.

```
ROLLBACK [TRANSACTION | WORK]
```

Remarks

Transactions are not started automatically. To start a transaction, you must do so explicitly using BEGIN TRANSACTION.

Transactions can be nested up to five levels deep. To start a nested transaction, use BEGIN TRANSACTION within the context of an existing transaction.

Transactions are not supported for linked tables.

TRANSFORM Statement

Creates a [crosstab query](#) .

Syntax

```
TRANSFORM aggfunction  
selectstatement  
PIVOT pivotfield [IN (value1[, value2[, ...]])]
```

The TRANSFORM statement has these parts:

Part	Description
<i>aggfunction</i>	An SQL aggregate function that operates on the selected data.
<i>selectstatement</i>	A SELECT statement.
<i>pivotfield</i>	The field or expression you want to use to create column headings in the query's result set.
<i>value1, value2</i>	Fixed values used to create column headings.

Remarks

When you summarize data using a crosstab query, you select values from specified fields or expressions as column headings so you can view data in a more compact format than with a [select query](#) .

TRANSFORM is optional but when included is the first statement in an [SQL string](#) . It precedes a SELECT statement that specifies the fields used as row headings and a GROUP BY clause that specifies row grouping. Optionally, you can include other clauses, such as WHERE , that specify additional selection or sorting criteria. You can also use [subqueries](#) as predicates— specifically, those in the WHERE clause— in a crosstab query.

The values returned in *pivotfield* are used as column headings in the query's result set. For example, pivoting the sales figures on the month of the sale in a crosstab query would create 12 columns. You can restrict *pivotfield* to create

headings from fixed values (*value1*, *value2*) listed in the optional IN clause. You can also include fixed values for which no data exists to create additional columns.

See Also

[FROM Clause](#)

[SELECT Statement](#)

[GROUP BY Clause](#)

[SQL Aggregate Functions](#)

[INNER JOIN Operation](#)

[SQL Subqueries](#)

[ORDER BY Clause](#)

[WHERE Clause](#)

INNER JOIN Operation

Combines records from two tables whenever there are matching values in a common field.

Syntax

```
FROM table1 INNER JOIN table2 ON table1.field1 compopr table2.field2
```

The INNER JOIN operation has these parts:

Part	Description
<i>table1</i> , <i>table2</i>	The names of the tables from which records are combined.
<i>field1</i> , <i>field2</i>	The names of the fields that are joined. If they are not numeric, the fields must be of the same data type and contain the same kind of data, but they do not have to have the same name.
<i>compopr</i>	Any relational comparison operator: "=", "<," ">," "<=," ">=," or "<>."

Remarks

You can use an INNER JOIN operation in any FROM clause. This is the most common type of join. Inner joins combine records from two tables whenever there are matching values in a field common to both tables.

You can use INNER JOIN with the Departments and Employees tables to select all the employees in each department. In contrast, to select all departments (even if some have no employees assigned to them) or all employees (even if some are not assigned to a department), you can use a LEFT JOIN or RIGHT JOIN operation to create an [outer join](#) .

If you try to join fields containing [Memo](#) or [OLE Object](#) data, an error occurs.

You can join any two numeric fields of like types. For example, you can join on [AutoNumber](#) and [Long](#) fields because they are like types. However, you cannot

join [Single](#) and [Double](#) types of fields.

The following example shows how you could join the Categories and Products tables on the CategoryID field:

```
SELECT CategoryName, ProductName  
  
FROM Categories INNER JOIN Products  
  
ON Categories.CategoryID = Products.CategoryID;
```

In the preceding example, CategoryID is the joined field, but it is not included in the query output because it is not included in the SELECT statement. To include the joined field, include the field name in the SELECT statement—in this case, Categories.CategoryID.

You can also link several ON clauses in a JOIN statement, using the following syntax:

```
SELECT fields  
FROM table1 INNER JOIN table2  
ON table1.field1 compopr table2.field1 AND  
ON table1.field2 compopr table2.field2) OR  
ON table1.field3 compopr table2.field3);
```

You can also nest JOIN statements using the following syntax:

```
SELECT fields  
FROM table1 INNER JOIN  
(table2 INNER JOIN [( table3  
[INNER JOIN [( tablex [INNER JOIN ...])]  
ON table3.field3 compopr tablex.fieldx]  
ON table2.field2 compopr table3.field3)  
ON table1.field1 compopr table2.field2;
```

A LEFT JOIN or a RIGHT JOIN may be nested inside an INNER JOIN, but an INNER JOIN may not be nested inside a LEFT JOIN or a RIGHT JOIN.

See Also

[FROM Clause](#)

[TRANSFORM Statement](#)

LEFT JOIN, RIGHT JOIN Operations UNION Operation
SELECT Statement

LEFT JOIN, RIGHT JOIN

Operations

Combines source-table records when used in any FROM clause.

Syntax

```
FROM table1 [ LEFT | RIGHT ] JOIN table2  
ON table1.field1 compopr table2.field2
```

The LEFT JOIN and RIGHT JOIN operations have these parts:

Part	Description
<i>table1</i> , <i>table2</i>	The names of the tables from which records are combined.
<i>field1</i> , <i>field2</i>	The names of the fields that are joined. The fields must be of the same data type and contain the same kind of data, but they do not need to have the same name.
<i>compopr</i>	Any relational comparison operator: "=", "<," ">," "<=," ">=," or "<>."

Remarks

Use a LEFT JOIN operation to create a [left outer join](#) . Left outer joins include all of the records from the first (left) of two tables, even if there are no matching values for records in the second (right) table.

Use a RIGHT JOIN operation to create a [right outer join](#) . Right outer joins include all of the records from the second (right) of two tables, even if there are no matching values for records in the first (left) table.

For example, you could use LEFT JOIN with the Departments (left) and Employees (right) tables to select all departments, including those that have no employees assigned to them. To select all employees, including those who are

not assigned to a department, you would use RIGHT JOIN.

The following example shows how you could join the Categories and Products tables on the CategoryID field. The query produces a list of all categories, including those that contain no products:

```
SELECT CategoryName,  
  
ProductName  
  
FROM Categories LEFT JOIN Products  
  
ON Categories.CategoryID = Products.CategoryID;
```

In this example, CategoryID is the joined field, but it is not included in the query results because it is not included in the SELECT statement. To include the joined field, enter the field name in the SELECT statement— in this case, Categories.CategoryID.

Notes

To create a query that includes only records in which the data in the joined fields is the same, use an INNER JOIN operation.

- A LEFT JOIN or a RIGHT JOIN can be nested inside an INNER JOIN, but an INNER JOIN cannot be nested inside a LEFT JOIN or a RIGHT JOIN. See the discussion of nesting in the INNER JOIN topic to see how to nest joins within other joins.
- You can link multiple ON clauses. See the discussion of clause linking in the INNER JOIN topic to see how this is done.

If you try to join fields containing [Memo](#) or [OLE Object](#) data, an error occurs.

See Also

[FROM Clause](#) [UNION Operation](#)
[INNER JOIN Operation](#)

UNION Operation

Creates a [union query](#), which combines the results of two or more independent queries or tables.

Syntax

```
[TABLE] query1 UNION [ALL] [TABLE] query2 [UNION [ALL] [TABLE] queryn [ ... ]]
```

The UNION operation has these parts:

Part	Description
<i>query1</i> - <i>n</i>	A SELECT statement, the name of a stored query, or the name of a stored table preceded by the TABLE keyword.

Remarks

You can merge the results of two or more queries, tables, and SELECT statements, in any combination, in a single UNION operation. The following example merges an existing table named New Accounts and a SELECT statement:

```
TABLE [New Accounts] UNION ALL  
  
SELECT *  
  
FROM Customers  
  
WHERE OrderAmount > 1000;
```

By default, no duplicate records are returned when you use a UNION operation; however, you can include the ALL predicate to ensure that all records are returned. This also makes the query run faster.

All queries in a UNION operation must request the same number of fields; however, the fields do not have to be of the same size or [data type](#).

Use [aliases](#) only in the first SELECT statement because they are ignored in any others. In the ORDER BY clause, refer to fields by what they are called in the first SELECT statement.

Notes

- You can use a GROUP BY or HAVING clause in each *query* argument to group the returned data.
- You can use an ORDER BY clause at the end of the last *query* argument to display the returned data in a specified order.

See Also

[ALL, DISTINCT, DISTINCTROW, TOP Predicates](#) [ORDER BY Clause](#)
[GROUP BY Clause](#) [SELECT Statement](#)
[HAVING Clause](#) [SQL Subqueries](#)
[INNER JOIN Operation](#) [WHERE Clause](#)
[LEFT JOIN, RIGHT JOIN Operations](#)

PARAMETERS Declaration

Declares the name and data type of each [parameter](#) in a [parameter query](#) .

Syntax

```
PARAMETERS name datatype [, name datatype [, ...]]
```

The PARAMETERS declaration has these parts:

Part	Description
<i>name</i>	The name of the parameter. Assigned to the Name property of the Parameter object and used to identify this parameter in the Parameters collection. You can use <i>name</i> as a string that is displayed in a dialog box while your application runs the query. Use brackets ([]) to enclose text that contains spaces or punctuation. For example, [Low price] and [Begin report with which month?] are valid <i>name</i> arguments.
<i>datatype</i>	One of the primary Microsoft Jet SQL data types or their synonyms.

Remarks

For queries that you run regularly, you can use a PARAMETERS declaration to create a parameter query. A parameter query can help automate the process of changing query [criteria](#) . With a parameter query, your code will need to provide the parameters each time the query is run.

The PARAMETERS declaration is optional but when included precedes any other statement, including SELECT .

If the declaration includes more than one parameter, separate them with commas. The following example includes two parameters:

```
PARAMETERS [Low price] Currency, [Beginning date] DateTime;
```

You can use *name* but not *datatype* in a WHERE or HAVING clause. The following example expects two parameters to be provided and then applies the criteria to records in the Orders table:

```
PARAMETERS [Low price] Currency,  
[Beginning date] DateTime;  
SELECT OrderID, OrderAmount  
FROM Orders  
WHERE OrderAmount > [Low price]  
AND OrderDate >= [Beginning date];
```

See Also

[EXECUTE Statement](#)

[HAVING Clause](#)

[Microsoft Jet Database Engine SQL Data Types](#)

[SELECT Statement](#)

[WHERE Clause](#)

WITH OWNERACCESS OPTION Declaration

In a multiuser environment with a [security-enabled workgroup](#), use this declaration with a query to give the user who runs the query the same [permissions](#) as the query's owner.

Syntax

```
sqlstatement WITH OWNERACCESS OPTION
```

Remarks

The WITH OWNERACCESS OPTION declaration is optional.

The following example enables the user to view salary information (even if the user does not otherwise have permission to view the Payroll table), provided that the query's owner does have that permission:

```
SELECT LastName,  
  
FirstName, Salary  
  
FROM Employees  
  
ORDER BY LastName  
  
WITH OWNERACCESS OPTION;
```

If a user is otherwise prevented from creating or adding to a table, you can use WITH OWNERACCESS OPTION to enable the user to run a [make-table](#) or [append query](#).

If you want to enforce workgroup security settings and users' permissions, do not include the WITH OWNERACCESS OPTION declaration.

This option requires you to have access to the System.mdw file associated with the database. It is useful only in security-enabled multiuser implementations.

See Also

[SELECT Statement](#)

CONSTRAINT Clause

A [constraint](#) is similar to an [index](#), although it can also be used to establish a [relationship](#) with another table.

You use the CONSTRAINT clause in ALTER TABLE and CREATE TABLE statements to create or delete constraints. There are two types of CONSTRAINT clauses: one for creating a constraint on a single field and one for creating a constraint on more than one field.

Note The [Microsoft Jet database engine](#) does not support the use of CONSTRAINT, or any of the [data definition language \(DDL\)](#) statements, with non-Microsoft Jet databases. Use the [DAO Create methods](#) instead.

Syntax

Single-field constraint:

```
CONSTRAINT name {PRIMARY KEY | UNIQUE | NOT NULL |  
REFERENCES foreigntable [(foreignfield1, foreignfield2)]  
[ON UPDATE CASCADE | SET NULL]  
[ON DELETE CASCADE | SET NULL]}
```

Multiple-field constraint:

```
CONSTRAINT name  
{PRIMARY KEY (primary1 [, primary2 [, ...]]) |  
UNIQUE (unique1 [, unique2 [, ...]]) |  
NOT NULL (notnull1 [, notnull2 [, ...]]) |  
FOREIGN KEY [NO INDEX] (ref1 [, ref2 [, ...]]) REFERENCES foreigntable  
[(foreignfield1 [, foreignfield2 [, ...]])]  
[ON UPDATE CASCADE | SET NULL]  
[ON DELETE CASCADE | SET NULL]}
```

The CONSTRAINT clause has these parts:

Part	Description
------	-------------

<i>name</i>	The name of the constraint to be created.
<i>primary1,</i> <i>primary2</i>	The name of the field or fields to be designated the primary key .
<i>unique1,</i> <i>unique2</i>	The name of the field or fields to be designated as a unique key.
<i>notnull1,</i> <i>notnull2</i>	The name of the field or fields that are restricted to non- Null values.
<i>ref1, ref2</i>	The name of a foreign key field or fields that refer to fields in another table.
<i>foreigntable</i>	The name of the foreign table containing the field or fields specified by <i>foreignfield</i> .
<i>foreignfield1,</i> <i>foreignfield2</i>	The name of the field or fields in <i>foreigntable</i> specified by <i>ref1</i> , <i>ref2</i> . You can omit this clause if the referenced field is the primary key of <i>foreigntable</i> .

Remarks

You use the syntax for a single-field constraint in the field-definition clause of an ALTER TABLE or CREATE TABLE statement immediately following the specification of the field's data type.

You use the syntax for a multiple-field constraint whenever you use the [reserved word](#) CONSTRAINT outside a field-definition clause in an ALTER TABLE or CREATE TABLE statement.

Using CONSTRAINT you can designate a field as one of the following types of constraints:

- You can use the UNIQUE reserved word to designate a field as a unique key. This means that no two records in the table can have the same value in this field. You can constrain any field or list of fields as unique. If a multiple-field constraint is designated as a unique key, the combined values of all fields in the index must be unique, even if two or more records have the same value in just one of the fields.
- You can use the [PRIMARY KEY](#) reserved words to designate one field or set of fields in a table as a primary key. All values in the primary key must be unique and not **Null**, and there can be only one primary key for a table.

Note Do not set a PRIMARY KEY constraint on a table that already has a primary key; if you do, an error occurs.

- You can use the [FOREIGN KEY](#) reserved words to designate a field as a foreign key. If the foreign table's primary key consists of more than one field, you must use a multiple-field constraint definition, listing all of the referencing fields, the name of the foreign table, and the names of the referenced fields in the foreign table in the same order that the referencing fields are listed. If the referenced field or fields are the foreign table's primary key, you do not have to specify the referenced fields. By default the database engine behaves as if the foreign table's primary key is the referenced fields.

Foreign key constraints define specific actions to be performed when a corresponding primary key value is changed:

- You can specify actions to be performed on the foreign table based on a corresponding action performed on a primary key in the table on which the CONSTRAINT is defined. For example, consider the following definition for the table Customers:

```
CREATE TABLE Customers (CustId INTEGER PRIMARY KEY, CLstNm
NCHAR VARYING (50))
```

Consider the following definition of the table Orders, which defines a foreign key relationship referencing the primary key of the Customers table:

```
CREATE TABLE Orders (OrderId INTEGER PRIMARY KEY, CustId
INTEGER, OrderNotes NCHAR VARYING (255), CONSTRAINT
FKOrdersCustId FOREIGN KEY (CustId) REFERENCES Customers ON
UPDATE CASCADE ON DELETE CASCADE
```

Both an ON UPDATE CASCADE and an ON DELETE CASCADE clause are defined on the foreign key. The ON UPDATE CASCADE clause means that if a customer's identifier (CustId) is updated in the Customer table, the update will be cascaded through the Orders table. Each order containing a corresponding customer identifier value will be updated automatically with the new value. The ON DELETE CASCADE clause means that if a customer is deleted from the Customer table, all rows in the Orders table containing the same customer identifier value will also be deleted.

Consider the following different definition of the table Orders, using the SET NULL action instead of the CASCADE action:

```
CREATE TABLE Orders (OrderId INTEGER PRIMARY KEY, CustId
INTEGER, OrderNotes NCHAR VARYING (255), CONSTRAINT
FKOrdersCustId FOREIGN KEY (CustId) REFERENCES Customers ON
UPDATE SET NULL ON DELETE SET NULL
```

The ON UPDATE SET NULL clause means that if a customer's identifier (CustId) is updated in the Customer table, the corresponding foreign key values in the Orders table will automatically be set to NULL. Similarly, the ON DELETE SET NULL clause means that if a customer is deleted from the Customer table, all corresponding foreign keys in the Orders table will automatically be set to NULL.

To prevent the automatic creation of indexes for foreign keys, the modifier NO INDEX can be used. This form of foreign key definition should be used only in cases where the resulting index values would be frequently duplicated. Where the values in a foreign key index are frequently duplicated, using an index can be less efficient than simply performing a table scan. Maintaining this type of index, with rows inserted and deleted from the table, degrades performance and does not provide any benefit.

See Also

[ADD USER Statement](#)

[ALTER USER or DATABASE Statement](#)

[ALTER TABLE Statement](#)

[CREATE INDEX Statement](#)

[CREATE PROCEDURE Statement](#)

[CREATE TABLE Statement](#)

[CREATE USER or GROUP Statement](#)

[CREATE VIEW Statement](#)

[DROP Statement](#)

[DROP USER or GROUP Statement](#)

[GRANT Statement](#)

[REVOKE Statement](#)

FROM Clause

Specifies the tables or queries that contain the fields listed in the SELECT statement.

Syntax

```
SELECT fieldlist  
FROM tableexpression [IN externaldatabase]
```

A SELECT statement containing a FROM clause has these parts:

Part	Description
<i>fieldlist</i>	The name of the field or fields to be retrieved along with any field-name aliases , SQL aggregate functions , selection predicates (ALL, DISTINCT, DISTINCTROW, or TOP), or other SELECT statement options.
<i>tableexpression</i>	An expression that identifies one or more tables from which data is retrieved. The expression can be a single table name, a saved query name, or a compound resulting from an INNER JOIN, LEFT JOIN, or RIGHT JOIN.
<i>externaldatabase</i>	The full path of an external database containing all the tables in <i>tableexpression</i> .

Remarks

FROM is required and follows any SELECT statement.

The order of the table names in *tableexpression* is not important.

For improved performance and ease of use, it is recommended that you use a [linked table](#) instead of an IN clause to retrieve data from an external database.

The following example shows how you can retrieve data from the Employees table:

```
SELECT LastName, FirstName  
FROM Employees;
```

See Also

[ALL DISTINCT, DISTINCTROW, TOP Predicates](#) [SELECT Statement](#)
[IN Clause](#) [SQL Aggregate Functions](#)
[LEFT JOIN, RIGHT JOIN Operations](#) [WHERE Clause](#)
[INNER JOIN Operation](#)

GROUP BY Clause

Combines records with identical values in the specified field list into a single record. A summary value is created for each record if you include an SQL aggregate function , such as **Sum** or **Count**, in the SELECT statement.

Syntax

```
SELECT fieldlist  
FROM table  
WHERE criteria  
[GROUP BY groupfieldlist]
```

A SELECT statement containing a GROUP BY clause has these parts:

Part	Description
<i>fieldlist</i>	The name of the field or fields to be retrieved along with any field-name aliases , SQL aggregate functions, selection predicates (ALL, DISTINCT, DISTINCTROW, or TOP), or other SELECT statement options.
<i>table</i>	The name of the table from which records are retrieved.
<i>criteria</i>	Selection criteria. If the statement includes a WHERE clause, the Microsoft Jet database engine groups values after applying the WHERE conditions to the records.
<i>groupfieldlist</i>	The names of up to 10 fields used to group records. The order of the field names in <i>groupfieldlist</i> determines the grouping levels from the highest to the lowest level of grouping.

Remarks

GROUP BY is optional.

Summary values are omitted if there is no SQL aggregate function in the SELECT statement.

Null values in GROUP BY fields are grouped and are not omitted. However, **Null** values are not evaluated in any SQL aggregate function.

Use the WHERE clause to exclude rows you do not want grouped, and use the HAVING clause to filter records after they have been grouped.

Unless it contains [Memo](#) or [OLE Object](#) data, a field in the GROUP BY field list can refer to any field in any table listed in the FROM clause, even if the field is not included in the SELECT statement, provided the SELECT statement includes at least one SQL aggregate function. The Microsoft® Jet database engine cannot group on Memo or OLE Object fields.

All fields in the SELECT field list must either be included in the GROUP BY clause or be included as arguments to an SQL aggregate function.

See Also

[ALL DISTINCT, DISTINCTROW, TOP Predicates](#) [SELECT](#)

[FROM Clause](#)

[HAVING Clause](#)

[ORDER BY Clause](#)

[SELECT...INTO Statement](#)

[SQL Aggregate Functions](#)

[WHERE Clause](#)

HAVING Clause

Specifies which grouped records are displayed in a SELECT statement with a GROUP BY clause. After GROUP BY combines records, HAVING displays any records grouped by the GROUP BY clause that satisfy the conditions of the HAVING clause.

Syntax

```
SELECT fieldlist
FROM table
WHERE selectcriteria
GROUP BY groupfieldlist
[HAVING groupcriteria]
```

A SELECT statement containing a HAVING clause has these parts:

Part	Description
<i>fieldlist</i>	The name of the field or fields to be retrieved along with any field-name aliases , SQL aggregate functions, selection predicates (ALL, DISTINCT, DISTINCTROW, or TOP), or other SELECT statement options.
<i>table</i>	The name of the table from which records are retrieved.
<i>selectcriteria</i>	Selection criteria. If the statement includes a WHERE clause, the Microsoft Jet database engine groups values after applying the WHERE conditions to the records.
<i>groupfieldlist</i>	The names of up to 10 fields used to group records. The order of the field names in <i>groupfieldlist</i> determines the grouping levels from the highest to the lowest level of grouping.
<i>groupcriteria</i>	An expression that determines which grouped records to display.

Remarks

HAVING is optional.

HAVING is similar to WHERE, which determines which records are selected. After records are grouped with GROUP BY, HAVING determines which records are displayed:

```
SELECT CategoryID,  
Sum(UnitsInStock)  
FROM Products  
GROUP BY CategoryID  
HAVING Sum(UnitsInStock) > 100 And Like "BOS*";
```

A HAVING clause can contain up to 40 expressions linked by logical operators, such as **And** and **Or**.

See Also

[ALL, DISTINCT, DISTINCTROW, or TOP Predicates](#)

[FROM Clause](#)

[GROUP BY Clause](#)

[ORDER BY Clause](#)

[SELECT Statement](#)

[SELECT..INTO Statement](#)

[SQL aggregate functions](#)

[WHERE Clause](#)

IN Clause

Identifies tables in any [external database](#) to which the [Microsoft Jet database engine](#) can connect, such as a dBASE or Paradox database or an external Microsoft® Jet database.

Syntax

To identify a destination table:

```
[SELECT | INSERT] INTO destination IN  
{path | ["path" "type"] | ["" [type; DATABASE = path]]}
```

To identify a source table:

```
FROM tableexpression IN  
{path | ["path" "type"] | ["" [type; DATABASE = path]]}
```

A SELECT statement containing an IN clause has these parts:

Part	Description
<i>destination</i>	The name of the external table into which data is inserted.
<i>tableexpression</i>	The name of the table or tables from which data is retrieved. This argument can be a single table name, a saved query, or a compound resulting from an INNER JOIN, LEFT JOIN, or RIGHT JOIN.
<i>path</i>	The full path for the directory or file containing <i>table</i> .
<i>type</i>	The name of the database type used to create <i>table</i> if a database is not a Microsoft Jet database (for example, dBASE III, dBASE IV, Paradox 3.x, or Paradox 4.x).

Remarks

You can use IN to connect to only one [external database](#) at a time.

In some cases, the *path* argument refers to the directory containing the database files. For example, when working with dBASE, Microsoft FoxPro®, or Paradox database tables, the *path* argument specifies the directory containing .dbf or .db files. The table file name is derived from the *destination* or *tableexpression* argument.

To specify a non-Microsoft Jet database, append a semicolon (;) to the name, and enclose it in single (' ') or double (" ") quotation marks. For example, either 'dBASE IV;' or "dBASE IV;" is acceptable.

You can also use the DATABASE reserved word to specify the external database. For example, the following lines specify the same table:

```
... FROM Table IN "" [dBASE IV; DATABASE=C:\DBASE\DATA\SALES;];
```

```
... FROM Table IN "C:\DBASE\DATA\SALES" "dBASE IV;"
```

Notes

For improved performance and ease of use, use a [linked table](#) instead of IN.

You can also use the IN reserved word as a comparison operator in an [expression](#).

See Also

[FROM Clause](#)

[INNER JOIN Clause](#)

[INSERT INTO Statement](#)

[LEFT JOIN, RIGHT JOIN Operations](#) [IN operator](#)

[SELECT Statement](#)

[SELECT...INTO Statement](#)

[SQL aggregate functions](#)

ORDER BY Clause

Sorts a query's resulting records on a specified field or fields in ascending or descending order.

Syntax

```
SELECT fieldlist
FROM table
WHERE selectcriteria
[ORDER BY field1 [ASC | DESC ][, field2 [ASC | DESC ]][, ...]]
```

A SELECT statement containing an ORDER BY clause has these parts:

Part	Description
<i>fieldlist</i>	The name of the field or fields to be retrieved along with any field-name aliases , SQL aggregate functions , selection predicates (ALL, DISTINCT, DISTINCTROW, or TOP), or other SELECT statement options.
<i>table</i>	The name of the table from which records are retrieved.
<i>selectcriteria</i>	Selection criteria. If the statement includes a WHERE clause, the Microsoft Jet database engine orders values after applying the WHERE conditions to the records.
<i>field1, field2</i>	The names of the fields on which to sort records.

Remarks

ORDER BY is optional. However, if you want your data displayed in sorted order, then you must use ORDER BY.

The default [sort order](#) is ascending (A to Z, 0 to 9). Both of the following examples sort employee names in last name order:

```
SELECT LastName, FirstName
FROM Employees
```

```
ORDER BY LastName;  
  
SELECT LastName, FirstName  
  
FROM Employees  
  
ORDER BY LastName ASC;
```

To sort in descending order (Z to A, 9 to 0), add the DESC reserved word to the end of each field you want to sort in descending order. The following example selects salaries and sorts them in descending order:

```
SELECT LastName, Salary  
  
FROM Employees  
  
ORDER BY Salary DESC, LastName;
```

If you specify a field containing [Memo](#) or [OLE Object](#) data in the ORDER BY clause, an error occurs. The Microsoft Jet database engine does not sort on fields of these types.

ORDER BY is usually the last item in an [SQL statement](#) .

You can include additional fields in the ORDER BY clause. Records are sorted first by the first field listed after ORDER BY. Records that have equal values in that field are then sorted by the value in the second field listed, and so on.

See Also

[ALL, DISTINCT, DISTINCTROW, or TOP Predicates](#)

[FROM Clause](#)

[GROUP BY Clause](#)

[HAVING Clause](#)

[SELECT Statement](#)

[SELECT...INTO Statement](#)

[SQL aggregate functions](#)

[WHERE Clause](#)

PROCEDURE Clause

Defines a name and optional [parameters](#) for a query.

Note The PROCEDURE clause has been superseded by the PROCEDURE statement. Although the PROCEDURE clause is still supported, the PROCEDURE statement provides a superset of the capability of the PROCEDURE clause and is the recommended syntax.

Syntax

```
PROCEDURE name [param1 datatype[, param2 datatype[, ...]]
```

The PROCEDURE clause has these parts:

Part	Description
<i>name</i>	A name for the procedure. It must follow standard naming conventions .
<i>param1</i> , <i>param2</i>	One or more field names or parameters . For example: <pre>PROCEDURE Sales_By_Country [Beginning Date] DateTime, [Ending Date] DateTime;</pre>
<i>datatype</i>	One of the primary Microsoft Jet SQL data types or their synonyms.

Remarks

An SQL procedure consists of a PROCEDURE clause (which specifies the name of the procedure), an optional list of parameter definitions, and a single [SQL statement](#) . For example, the procedure Get_Part_Number might run a query that retrieves a specified part number.

Notes

- If the clause includes more than one field definition (that is, *param-datatype* pairs), separate them with commas.

- The PROCEDURE clause must be followed by an SQL statement (for example, a SELECT or UPDATE statement).

See Also

[DELETE Statement](#)

[PARAMETERS Declaration](#)

[EXECUTE Statement](#)

[SELECT statement](#)

[Microsoft Jet Database Engine SQL Data Types](#) [UPDATE statement](#)

WHERE Clause

Specifies which records from the tables listed in the FROM clause are affected by a SELECT , UPDATE , or DELETE statement.

Syntax

```
SELECT fieldlist  
FROM tableexpression  
WHERE criteria
```

A SELECT statement containing a WHERE clause has these parts:

Part	Description
<i>fieldlist</i>	The name of the field or fields to be retrieved along with any field-name aliases , selection predicates (ALL, DISTINCT, DISTINCTROW, or TOP), or other SELECT statement options.
<i>tableexpression</i>	The name of the table or tables from which data is retrieved.
<i>criteria</i>	An expression that records must satisfy to be included in the query results.

Remarks

The [Microsoft Jet database engine](#) selects the records that meet the conditions listed in the WHERE clause. If you do not specify a WHERE clause, your query returns all rows from the table. If you specify more than one table in your query and you have not included a WHERE clause or a JOIN clause, your query generates a [Cartesian product](#) of the tables.

WHERE is optional, but when included, follows FROM. For example, you can select all employees in the sales department (WHERE Dept = 'Sales') or all customers between the ages of 18 and 30 (WHERE Age Between 18 And 30).

If you do not use a JOIN clause to perform SQL join operations on multiple

tables, the resulting **Recordset** object will not be updatable.

WHERE is similar to HAVING. WHERE determines which records are selected. Similarly, once records are grouped with GROUP BY, HAVING determines which records are displayed.

Use the WHERE clause to eliminate records you do not want grouped by a GROUP BY clause.

Use various expressions to determine which records the SQL statement returns. For example, the following SQL statement selects all employees whose salaries are more than \$21,000:

```
SELECT LastName, Salary
FROM Employees
WHERE Salary > 21000;
```

A WHERE clause can contain up to 40 expressions linked by logical operators, such as **And** and **Or**.

When you enter a field name that contains a space or punctuation, surround the name with brackets ([]). For example, a customer information table might include information about specific customers :

```
SELECT [Customer's Favorite Restarant]
```

When you specify the *criteria* argument, [date literals](#) must be in U.S. format, even if you are not using the U.S. version of the Microsoft® Jet database engine. For example, May 10, 1996, is written 10/5/96 in the United Kingdom and 5/10/96 in the United States. Be sure to enclose your date literals with the number sign (#) as shown in the following examples.

To find records dated May 10, 1996 in a United Kingdom database, you must use the following SQL statement:

```
SELECT *
FROM Orders
WHERE ShippedDate = #5/10/96#;
```

You can also use the **DateValue** function which is aware of the international settings established by Microsoft Windows®. For example, use this code for the United States:

```
SELECT *  
  
FROM Orders  
  
WHERE ShippedDate = DateValue('5/10/96');
```

And use this code for the United Kingdom:

```
SELECT *  
  
FROM Orders  
  
WHERE ShippedDate = DateValue('10/5/96');
```

Note If the column referenced in the criteria string is of type [GUID](#), the criteria expression uses a slightly different syntax:

```
WHERE ReplicaID = {GUID {12345678-90AB-CDEF-1234-567890ABCDEF}}
```

Be sure to include the nested braces and hyphens as shown.

See Also

[ALL, DISTINCT, DISTINCTROW, or TOP
Predicates](#)

[DELETE Statement](#)

[FROM Statement](#)

[GROUP BY](#)

[HAVING Clause](#)

[IN Clause](#)

[INNER JOIN Operation](#)

[LEFT JOIN, RIGHT JOIN
Operations](#)

[ORDER BY Clause](#)

[SELECT Statement](#)

[SELECT...INTO Statement](#)

[SQL aggregate functions](#)

[UPDATE Statement](#)

Avg Function

Calculates the arithmetic mean of a set of values contained in a specified field on a query.

Syntax

Avg(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the numeric data you want to average or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

The average calculated by **Avg** is the arithmetic mean (the sum of the values divided by the number of values). You could use **Avg**, for example, to calculate average freight cost.

The **Avg** function does not include any [Null](#) fields in the calculation.

You can use **Avg** in a query expression and in the [SQL](#) property of a [QueryDef](#) object or when creating a [Recordset](#) object based on an SQL query.

See Also

[Calculating Fields in SQL Functions](#) [SQL aggregate functions](#)

Count Function

Calculates the number of records returned by a query.

Syntax

Count(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the data you want to count or an expression that performs a calculation using the data in the field. Operands in *expr* can include the name of a table field or function (which can be either intrinsic or user-defined but not other SQL aggregate functions). You can count any kind of data, including text.

Remarks

You can use **Count** to count the number of records in an underlying query. For example, you could use **Count** to count the number of orders shipped to a particular country.

Although *expr* can perform a calculation on a field, **Count** simply tallies the number of records. It does not matter what values are stored in the records.

The **Count** function does not count records that have [Null](#) fields unless *expr* is the asterisk (*) [wildcard character](#) . If you use an asterisk, **Count** calculates the total number of records, including those that contain **Null** fields. **Count(*)** is considerably faster than **Count([Column Name])**. Do not enclose the asterisk in quotation marks (' '). The following example calculates the number of records in the Orders table:

```
SELECT Count(*)
```

```
AS TotalOrders FROM Orders;
```

If *expr* identifies multiple fields, the **Count** function counts a record only if at least one of the fields is not **Null**. If all of the specified fields are **Null**, the record is not counted. Separate the field names with an ampersand (&). The following example shows how you can limit the count to records in which either

ShippedDate or Freight is not **Null**:

```
SELECT
```

```
Count('ShippedDate & Freight')
```

```
AS [Not Null] FROM Orders;
```

You can use **Count** in a query expression. You can also use this expression in the [SQL](#) property of a [QueryDef](#) object or when creating a [Recordset](#) object based on an SQL query.

See Also

[SELECT Statement](#) [SUM Function](#)

[SQL aggregate functions](#)

First, Last Functions

Return a field value from the first or last record in the result set returned by a query.

Syntax

First(*expr*)

Last(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the data you want to use or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

The **First** and **Last**[MoveFirst](#) and [MoveLast](#) methods of a [DAO Recordset](#) object. They simply return the value of a specified field in the first or last record, respectively, of the result set returned by a query. Because records are usually returned in no particular order (unless the query includes an ORDER BY clause), the records returned by these functions will be arbitrary.

See Also

[Calculating Fields in SQL Functions](#) [SQL aggregate functions](#)
[ORDER BY clause](#)

Min, Max Functions

Return the minimum or maximum of a set of values contained in a specified field on a query.

Syntax

Min(*expr*)

Max(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the data you want to evaluate or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

You can use **Min** and **Max** to determine the smallest and largest values in a field based on the specified aggregation, or grouping. For example, you could use these functions to return the lowest and highest freight cost. If there is no aggregation specified, then the entire table is used.

You can use **Min** and **Max** in a query expression and in the [SQL](#) property of a [QueryDef](#) object or when creating a [Recordset](#) object based on an SQL query.

See Also

[SQL aggregate functions](#)

StDev, StDevP Functions

Return estimates of the [standard deviation](#) for a population or a population sample represented as a set of values contained in a specified field on a query.

Syntax

StDev(*expr*)

StDevP(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the numeric data you want to evaluate or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

The **StDevP** function evaluates a population, and the **StDev** function evaluates a population sample.

If the underlying query contains fewer than two records (or no records, for the **StDevP** function), these functions return a **Null** value (which indicates that a standard deviation cannot be calculated).

You can use the **StDev** and **StDevP** functions in a query expression. You can also use this expression in the [SQL](#) property of a [QueryDef](#) object or when creating a [Recordset](#) object based on an SQL query.

See Also

[Avg Function](#) [SQL aggregate functions](#)

[SELECT Statement SUM Function](#)

Sum Function

Returns the sum of a set of values contained in a specified field on a query.

Syntax

Sum(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the numeric data you want to add or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

The **Sum** function totals the values in a field. For example, you could use the **Sum** function to determine the total cost of freight charges.

The **Sum** function ignores records that contain [Null](#) fields. The following example shows how you can calculate the sum of the products of UnitPrice and Quantity fields:

```
SELECT  
  
Sum(UnitPrice * Quantity)  
  
AS [Total Revenue] FROM [Order Details];
```

You can use the **Sum** function in a query expression. You can also use this expression in the [SQL](#) property of a [QueryDef](#) object or when creating a [Recordset](#) based on an SQL query.

See Also

[Count Function](#) [SQL aggregate functions](#)
[SELECT Statement](#)

Var, VarP Functions

Return estimates of the [variance](#) for a population or a population sample represented as a set of values contained in a specified field on a query.

Syntax

Var(*expr*)

VarP(*expr*)

The *expr* placeholder represents a [string expression](#) identifying the field that contains the numeric data you want to evaluate or an expression that performs a calculation using the data in that field. Operands in *expr* can include the name of a table field, a constant, or a function (which can be either intrinsic or user-defined but not one of the other SQL aggregate functions).

Remarks

The **VarP** function evaluates a population, and the **Var** function evaluates a population sample.

If the underlying query contains fewer than two records, the **Var** and **VarP** functions return a [Null](#) value, which indicates that a variance cannot be calculated.

You can use the **Var** and **VarP** functions in a query expression or in an [SQL statement](#) .

See Also

[SQL aggregate functions](#)

ALL, DISTINCT, DISTINCTROW, TOP Predicates

Specifies records selected with [SQL](#) queries.

Syntax

```
SELECT [ALL | DISTINCT | DISTINCTROW | [TOP n [PERCENT]]]  
FROM table
```

A SELECT statement containing these predicates has the following parts:

Part	Description
	Assumed if you do not include one of the predicates. The Microsoft Jet database engine selects all of the records that meet the conditions in the SQL statement . The following two examples are equivalent and return all records from the Employees table:
ALL	<pre>SELECT ALL * FROM Employees ORDER BY EmployeeID; SELECT * FROM Employees ORDER BY EmployeeID;</pre>
	Omits records that contain duplicate data in the selected fields. To be included in the results of the query, the values for each field listed in the SELECT statement must be unique. For example, several employees listed in an Employees table may have the same last name. If two records contain Smith in the LastName field, the following SQL statement returns only one record that contains Smith:
	<pre>SELECT DISTINCT LastName</pre>

DISTINCT FROM Employees;

If you omit **DISTINCT**, this query returns both Smith records.

If the **SELECT** clause contains more than one field, the combination of values from all fields must be unique for a given record to be included in the results.

The output of a query that uses **DISTINCT** is not updatable and does not reflect subsequent changes made by other users.

Omits data based on entire duplicate records, not just duplicate fields. For example, you could create a query that joins the Customers and Orders tables on the CustomerID field. The Customers table contains no duplicate CustomerID fields, but the Orders table does because each customer can have many orders. The following SQL statement shows how you can use **DISTINCTROW** to produce a list of companies that have at least one order but without any details about those orders:

DISTINCTROW SELECT DISTINCTROW CompanyName
FROM Customers INNER JOIN Orders
ON Customers.CustomerID = Orders.CustomerID
ORDER BY CompanyName;

If you omit **DISTINCTROW**, this query produces multiple rows for each company that has more than one order.

DISTINCTROW has an effect only when you select fields from some, but not all, of the tables used in the query. **DISTINCTROW** is ignored if your query includes only one table, or if you output fields from all tables.

Returns a certain number of records that fall at the top or the bottom of a range specified by an **ORDER BY** clause. Suppose you want the names of the top 25 students from the class of 1994:

```
SELECT TOP 25  
FirstName, LastName
```

```
FROM Students
WHERE GraduationYear = 1994
ORDER BY GradePointAverage DESC;
```

If you do not include the ORDER BY clause, the query will return an arbitrary set of 25 records from the Students table that satisfy the WHERE clause.

TOP *n*
[PERCENT]

The TOP predicate does not choose between equal values. In the preceding example, if the twenty-fifth and twenty-sixth highest grade point averages are the same, the query will return 26 records.

You can also use the PERCENT reserved word to return a certain percentage of records that fall at the top or the bottom of a range specified by an ORDER BY clause. Suppose that, instead of the top 25 students, you want the bottom 10 percent of the class:

```
SELECT TOP 10 PERCENT
FirstName, LastName
FROM Students
WHERE GraduationYear = 1994
ORDER BY GradePointAverage ASC;
```

The ASC predicate specifies a return of bottom values. The value that follows TOP must be an unsigned [Integer](#) .

TOP does not affect whether or not the query is updatable.

table

The name of the table from which records are retrieved.

See Also

[FROM Clause SELECT Statement](#)

Between...And Operator

Determines whether the value of an expression falls within a specified range of values. You can use this operator within [SQL statements](#) .

Syntax

expr [**Not**] **Between** *value1* **And** *value2*

The **Between...And** operator syntax has these parts:

Part	Description
<i>expr</i>	Expression identifying the field that contains the data you want to evaluate.
<i>value1</i> , <i>value2</i>	Expressions against which you want to evaluate <i>expr</i> .

Remarks

If the value of *expr* is between *value1* and *value2* (inclusive), the **Between...And** operator returns **True**; otherwise, it returns **False**. You can include the **Not** logical operator to evaluate the opposite condition (that is, whether *expr* lies outside the range defined by *value1* and *value2*).

You might use **Between...And** to determine whether the value of a field falls within a specified numeric range. The following example determines whether an order was shipped to a location within a range of postal codes. If the postal code is between 98101 and 98199, the **IIf** function returns “Local”. Otherwise, it returns “Nonlocal”.

```
SELECT IIf(PostalCode Between 98101 And 98199, "Local", "Nonlocal")  
FROM Publishers
```

If *expr*, *value1*, or *value2* is [Null](#) , **Between...And** returns a **Null** value.

Because [wildcard characters](#) , such as *, are treated as literals, you cannot use

them with the **Between...And** operator. For example, you cannot use 980* and 989* to find all postal codes that start with 980 to 989. Instead, you have two alternatives for accomplishing this. You can add an expression to the query that takes the left three characters of the text field and use **Between...And** on those characters. Or you can pad the high and low values with extra characters— in this case, 98000 to 98999, or 98000 to 98999 – 9999 if using extended postal codes. (You must omit the – 0000 from the low values because otherwise 98000 is dropped if some postal codes have extended sections and others do not.)

See Also

[IN Clause](#) [SQL Expressions](#)
[WHERE Clause](#)

In Operator

Determines whether the value of an expression is equal to any of several values in a specified list.

Syntax

expr [**Not**] **In**(*value1*, *value2*, . . .)

Remarks

The **In** operator syntax has these parts:

Part	Description
<i>expr</i>	Expression identifying the field that contains the data you want to evaluate.
<i>value1</i> , <i>value2</i>	Expression or list of expressions against which you want to evaluate <i>expr</i> .

If *expr* is found in the list of values, the **In** operator returns **True**; otherwise, it returns **False**. You can include the **Not** logical operator to evaluate the opposite condition (that is, whether *expr* is not in the list of values).

For example, you can use **In** to determine which orders are shipped to a set of specified regions:

```
SELECT *  
  
FROM Orders  
  
WHERE ShipRegion In ('Avon', 'Glos', 'Som')
```

See Also

[SQL Expressions WHERE Clause](#)

Like Operator

Compares a [string expression](#) to a pattern in an [SQL](#) expression.

Syntax

expression **Like** "pattern"

The **Like** operator syntax has these parts:

Part	Description
<i>expression</i>	SQL expression used in a WHERE clause .
<i>pattern</i>	String or character string literal against which <i>expression</i> is compared.

Remarks

You can use the **Like** operator to find values in a field that match the pattern you specify. For *pattern*, you can specify the complete value (for example, Like "Smith"), or you can use [wildcard characters](#) to find a range of values (for example, Like "Sm*").

In an expression, you can use the **Like** operator to compare a field value to a string expression. For example, if you enter Like "C*" in an SQL query, the query returns all field values beginning with the letter C. In a [parameter query](#) , you can prompt the user for a pattern to search for.

The following example returns data that begins with the letter P followed by any letter between A and F and three digits:

```
Like "P[A-F]###"
```

The following table shows how you can use **Like** to test expressions for different patterns.

Kind of match	Pattern	Match (returns True)	No match (returns False)
---------------	---------	-------------------------	-----------------------------

Multiple characters	a*a	aa, aBa, aBBBa	aBC
	ab	abc, AABB, Xab	aZb, bac
Special character	a[*]a	a*a	aaa
Multiple characters	ab*	abcdefg, abc	cab, aab
Single character	a?a	aaa, a3a, aBa	aBBBa
Single digit	a#a	a0a, a1a, a2a	aaa, a10a
Range of characters	[a-z]	f, p, j	2, &
Outside a range	[!a-z]	9, &, %	b, a
Not a digit	[!0-9]	A, a, &, ~	0, 1, 9
Combined	a[!b-m]#	An9, az0, a99	abc, aj0

See Also

[SQL Expressions Using Wildcard Characters in String Comparisons WHERE Clause](#)

Customizing Windows Registry Settings for Microsoft Jet

If your application cannot work correctly with the default functionality of the [Microsoft Jet database engine](#) , you may have to change the settings in the Microsoft® Windows® Registry to suit your needs. The Windows Registry can also be used to tune the operation of the [installable ISAM](#) and [ODBC driver](#) .

You can customize the settings in the Windows Registry in four different ways:

[Using Regedit.exe to Overwrite the Default Settings](#)

[Creating a Portion in Your Application's Registry Tree to Manage the Settings](#)

[Using the SetOption Method from DAO](#)

[Using the Connection Properties in the Microsoft OLE DB Provider for Jet](#)

You can also edit the Windows Registry to specify the following:

- Settings used for interaction with Microsoft Excel, Lotus, Paradox, and dBASE databases. See [Initializing the Paradox Database Driver](#) , [Initializing the Microsoft Excel Driver](#) , [Initializing the Lotus Driver](#) , and [Initializing the dBASE Database Driver](#) .
- Settings used by Microsoft ODBC for interaction with SQL databases. See [Initializing the Microsoft ODBC Database Driver](#) .
- Settings that affect how the Microsoft Jet database engine reads and saves data. See [Initializing the Microsoft Jet 2.5 Database Engine Driver](#) , [Initializing the Microsoft Jet 3.5 Database Engine Driver](#) , and [Initializing the Microsoft Jet 4.0 Database Engine Driver](#) .
- Settings used to control how the Microsoft Jet database engine interacts with the Internet and Microsoft Exchange. See [Initializing the Text and HTML Data Source Driver](#) and [Initializing the Microsoft Exchange Data Source Driver](#) .
- Settings used to control how the Microsoft Jet database engine interacts with data imported as plain text. See [Initializing the Text and HTML Data](#)

[Source Driver](#) .

Creating a Portion in Your Application's Registry Tree to Manage the Settings

To customize the Microsoft® Windows® Registry settings, you can create a Microsoft Jet portion in your application's registry tree to manage the settings for the Microsoft Jet database engine. The easiest way to accomplish this is to export the existing Microsoft Jet key and then import it into your application's tree with the Regedit.exe Export and Import commands. You can then alter the values in your new registry tree. If you have supplied any values in the Engines subfolder, Microsoft Jet loads those settings when the application starts. Any values not entered in your client application's registry tree are loaded from shadow settings.

For your application to load the appropriate portion of the Windows Registry key you must specify the location with the DAO **INIPath** property. Your application must set the **INIPath** property before executing any other DAO code. The scope of this setting is limited to your application and cannot be changed without restarting your application.

Note Although creating a Microsoft Jet portion in your application's registry is more flexible than overwriting the Microsoft Jet default entries, it requires that you maintain the registry tree. Every time changes are required in the default settings, you will need to edit the Registry.

See Also

[Using Regedit.exe to Overwrite the Default Settings](#)

[Using the SetOption Method from DAO](#)

[Using the Connection Properties in the Microsoft OLE DB Provider for Jet](#)

Using Regedit.exe to Overwrite the Default Settings

To customize the Microsoft® Windows® Registry settings, you can use Regedit.exe to overwrite the default settings that are established when the Microsoft Jet database engine is registered. This method of modification is the least flexible method because all applications that do not specify another registry location will have these new default settings.

See Also

[Creating a Portion in Your Application's Registry Tree to Manage the Settings](#) [Using the Connection Properties in the Microsoft OLE DB Provider for Jet](#)
[Using the SetOption Method from DAO](#)

Initializing the dBASE Database Driver

When you install the dBASE database driver, the Setup program writes a set of default values to the Microsoft® Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly; use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM format settings for the dBASE database driver.

The Paradox database driver will work in one of two modes, depending upon whether the Borland Database Engine (BDE) is installed. Paradox data is only updateable with the BDE. Without the BDE, the Paradox data can be Read, Exported, or Linked to read-only.

dBASE Initialization Settings

The Jet\4.0\Engines\Xbase folder includes initialization settings for the msxbde40.dll driver, used for access to external dBASE data sources. Typical settings for the entries in this folder are shown in the following example.

```
win32=<path>\msxbde40.dll
```

```
NetworkAccess=0n
```

```
PageTimeout=600
```

```
INFPATH=C:\DBASE\SYSTEM
```

```
CollatingSequence=ASCII
```

```
DataCodePage=OEM
```

```
Deleted=0n
```

```
DbcsStr=0n
```

```
Century=0ff
```

Date=MDY

Mark=47

Exact=Off

The Microsoft Jet database engine uses the Xbase folder entries as follows.

Entry	Description
win32	The location of Msxbde40.dll. The full path is determined at the time of installation.
NetworkAccess	A binary indicator for file locking preference. If NetworkAccess is set to 00, tables are opened for exclusive access, regardless of the settings of the OpenDatabase and OpenRecordset methods' <i>exclusive</i> argument. The default value is 01.
PageTimeout	The length of time between when data is placed in an internal cache and when it is invalidated. The value is specified in 100 millisecond units. The default is 600 units or 60 seconds.
INFPPath	The full path to the .inf file directory. The Microsoft Jet database engine first looks for an .inf file in the directory containing the table. If the .inf file is not in the database directory, it looks in the INFPPath. If there is no INFPPath, it uses whatever index files (.cdx or .mdx) it finds in the database directory. This entry is not written by the installation procedure. This setting is only used if the BDE is not present on the machine.
CollatingSequence	The collating sequence for all dBASE tables created or opened using the Microsoft Jet database engine. Possible values are ASCII and International. The default is ASCII. This setting is only used if the BDE is not present on the machine.

An indicator of how text pages are stored. Possible settings are:

DataCodePage

- OEM— OemToAnsi and AnsiToOem conversions done.
- ANSI— OemToAnsi and AnsiToOem conversions not done.

The default is OEM.

Deleted

A binary indicator that determines how records marked for deletion are handled by the Microsoft Jet database engine. A value of 01 corresponds to the dBASE command SET DELETED ON and indicates never to retrieve or position on a deleted record. A value of 00 corresponds to the dBASE command SET DELETED OFF and indicates to treat a deleted record like any other record. The default is 00.

DbcsStr

A binary indicator that determines how the string functions will handle Far East character set data. A value of 01 corresponds to the dBASE command SET KANJISTRING ON and indicates that strings should be treated as a dbcs character stream. A value of 00 indicates that strings should be treated as a simple byte stream.

Century

A binary indicator for formatting the century component of dates in cases where date-to-string functions are used in index expressions. A value of 01 corresponds to the dBASE command SET CENTURY ON and a value of 00 corresponds to the dBASE command SET CENTURY OFF. The default is 00.

Date

The date formatting style to use in cases where date-to-string functions are used in index expressions. The possible settings for this entry, which corresponds to the dBASE SET DATE command, are American, ANSI, British, French, DMY, German, Italian, Japan, MDY, USA, and YMD. The default is MDY.

The decimal value of the ASCII character used to separate date parts. The default depends on the Date setting as follows:

- Mark
- "/" (American, MDY)
 - "." (ANSI)
 - "/" (British, French, DMY)
 - "." (German)
 - "-" (Italian)
 - "/" (Japan, YMD)
 - "-" (USA)

A value of 0 specifies that the system should use the separator usually associated with the selected date format.

Exact

A binary indicator for string comparisons. A value of 01 corresponds to the dBASE command SET EXACT ON. A value of 00 corresponds to the dBASE command SET EXACT OFF. The default is 00.

dBASE ISAM Formats

The Jet\4.0\ISAM Formats\dBASE III folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Xbase
ExportFilter	REG_SZ	dBASE III (*.dbf)
ImportFilter	REG_SZ	dBASE III (*.dbf)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	01
IndexFilter	REG_SZ	dBASE Index (*.ndx)
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file. Create a table in the current database that

ResultTextLink	REG_SZ	is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a dBASE III file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

The Jet\4.0\ISAM Formats\dBASE IV folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Xbase
ExportFilter	REG_SZ	dBASE IV (*.dbf)
ImportFilter	REG_SZ	dBASE IV (*.dbf)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	01
IndexFilter	REG_SZ	dBASE Index (*.ndx; *.mdx)
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a dBASE IV file. This process will overwrite the data if exported to an existing file.

SupportsLongNames REG_BINARY 00

The Jet\4.0\ISAM Formats\dBASE 5.x folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Xbase
ExportFilter	REG_SZ	dBASE 5 (*.dbf)
ImportFilter	REG_SZ	dBASE 5 (*.dbf)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	01
IndexFilter	REG_SZ	dBASE Index (*.ndx; *.mdx)
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a dBASE 5 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Lotus Driver

When you install the Lotus database driver, the Setup program writes a set of default values to the Microsoft® Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly; use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Lotus database driver.

Lotus Initialization Settings

The Jet\4.0\Engines\Lotus folder includes initialization settings for the msltus40.dll driver, used for external access to Lotus spreadsheets. Typical settings for the entries under this heading are shown in the following example.

```
win32=<path>\msltus40.dll
```

```
TypeGuessRows=8
```

```
ImportMixedTypes=Text
```

```
AppendBlankRows=4
```

```
FirstRowHasNames=Yes
```

The Microsoft Jet database engine uses the Lotus folder entries as follows.

Entry	Description
win32	The location of msltus40.dll. The full path is determined at the time of installation.
TypeGuessRows	The number of rows to be checked for the data type. The data type is determined based on the most frequently found data type in the selection. If there is a tie, the data type is determined in the following order: Number, Currency, Date, Text, Long Text. If data is encountered that does not match the data type guessed for the column, it is returned as a Null value. On import, if a column has mixed data types, the entire column will be converted according to the

ImportMixedTypes setting.

The default number of rows to be checked is 8.

ImportMixedTypes Can be set to MajorityType or Text. If set to MajorityType, columns of mixed data types will be cast to the predominate data type on import. If set to Text, columns of mixed data types will be cast to Text on import. The default is Text.

AppendBlankRows The number of blank rows to be appended to the end of a WK1 worksheet before new data is added. For example, if AppendBlankRows is set to 4, Microsoft Jet will append 4 blank rows to the end of the worksheet before appending rows that contain data. Integer values for this setting can range from 0 to 16; the default is 01 (one additional row appended).

FirstRowHasNames A binary value that indicates whether the first row of the table contains column names. A value of 01 indicates that, during import, column names are taken from the first row. A value of 00 indicates no column names in the first row; column names appear as F1, F2, F3, and so on. The default is 01.

Lotus ISAM Formats

The Jet\4.0\ISAM Formats\Lotus WK1 folder contains the following entries.

Entry name	Value
Engine	Lotus
ExportFilter	Lotus 1-2-3 WK1 (*.wk1)
ImportFilter	Lotus 1-2-3 (*.wk*)
CanLink	00
OneTablePerFile	00
IsamType	1
IndexDialog	00

CreateDBOnExport	01
ResultTextImport	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextExport	Export data from the current database into a Lotus 1-2-3 Version 2 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	01

The Jet\4.0\ISAM Formats\Lotus WK3 folder contains the following entries.

Entry name	Value
Engine	Lotus
ExportFilter	Lotus 1-2-3 WK3 (*.wk3)
CanLink	00
OneTablePerFile	00
IsamType	1
IndexDialog	00
CreateDBOnExport	01
ResultTextExport	Export data from the current database into a Lotus 1-2-3 Version 3 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	01

The Jet\4.0\ISAM Formats\Lotus WK4 folder contains the following entries.

Entry name	Value
Engine	Lotus
CanLink	00
OneTablePerFile	00

IsamType	1
IndexDialog	00
CreateDBOnExport	01
SupportsLongNames	01

The Jet\4.0\ISAM Formats\Lotus WJ2 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Lotus
ExportFilter	REG_SZ	Lotus 1-2-3 WJ2 (*.wj2)
ImportFilter	REG_SZ	Lotus 1-2-3/DOS (*.wj*)
CanLink	REG_BINARY	00
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	01
ResultTextExport	REG_SZ	Export data from the current database into a Lotus 1-2-3 Version 2 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	01

The Jet\4.0\ISAM Formats\Lotus WJ3 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Lotus
CanLink	REG_BINARY	00
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1
IndexDialog	REG_BINARY	00

CreateDBOnExport REG_BINARY 01
SupportsLongNames REG_BINARY 01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Microsoft Excel Driver

When you install the Microsoft® Excel driver, the Setup program writes a set of default values to the Microsoft Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly; use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Microsoft Excel database driver.

Microsoft Excel Initialization Settings

The Jet\4.0\Engines\Excel folder includes initialization settings for the msexcl40.dll driver, used for external access to Microsoft Excel worksheets. Typical settings for the entries in this folder are shown in the following example.

```
win32=<path>\ msexcl40.dll
```

```
TypeGuessRows=8
```

```
ImportMixedTypes=Text
```

```
AppendBlankRows=1
```

```
FirstRowHasNames=Yes
```

The Microsoft Jet database engine uses the Excel folder entries as follows.

Entry	Description
win32	The location of msexcl40.dll. The full path is determined at the time of installation.
TypeGuessRows	The number of rows to be checked for the data type. The data type is determined given the maximum number of kinds of data found. If there is a tie, the data type is determined in the following order: Number, Currency, Date, Text, Boolean. If data is encountered that does not match the data type guessed for the column, it is returned

as a **Null** value. On import, if a column has mixed data types, the entire column will be cast according to the ImportMixedTypes setting.

The default number of rows to be checked is 8.

ImportMixedTypes Can be set to MajorityType or Text. If set to MajorityType, columns of mixed data types will be cast to the predominate data type on import. If set to Text, columns of mixed data types will be cast to Text on import. The default is Text.

AppendBlankRows The number of blank rows to be appended to the end of a Version 3.5 or Version 4.0 worksheet before new data is added. For example, if AppendBlankRows is set to 4, Microsoft Jet will append 4 blank rows to the end of the worksheet before appending rows that contain data. Integer values for this setting can range from 0 to 16; the default is 01 (one additional row appended).

FirstRowHasNames A binary value that indicates whether the first row of the table contains column names. A value of 01 indicates that, during import, column names are taken from the first row. A value of 00 indicates no column names in the first row; column names appear as F1, F2, F3, and so on. The default is 01.

Microsoft Excel ISAM Formats

The Jet\4.0\ISAM Formats\Excel 3.0 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Excel
ExportFilter	REG_SZ	Microsoft Excel 3 (*.xls)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1

IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	01
ResultTextExport	REG_SZ	Export data from the current database into a Microsoft Excel 3.0 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	01

The Jet\4.0\ISAM Formats\Excel 4.0 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Excel
ExportFilter	REG_SZ	Microsoft Excel 4 (*.xls)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	01
ResultTextExport	REG_SZ	Export data from the current database into a Microsoft Excel 4.0 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	01

The Jet\4.0\ISAM Formats\Excel 5.0 folder contains the following entries, which apply to Microsoft Excel versions 5.0 and 7.0.

Entry name	Type	Value
Engine	REG_SZ	Excel
ExportFilter	REG_SZ	Microsoft Excel 5-7 (*.xls)

ImportFilter	REG_SZ	Microsoft Excel (*.xls)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	01
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a Microsoft Excel 5.0 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	01

The Jet\4.0\ISAM Formats\Excel 8.0 folder contains the following entries, which apply to Microsoft Excel 97.

Entry name	Type	Value
Engine	REG_SZ	Excel
ExportFilter	REG_SZ	Microsoft Excel 97-2000 (*.xls)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	1
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	01
		Export data from the current database into a Microsoft Excel 97 file. This

ResultTextExport REG_SZ process will overwrite the data if exported to an existing file.

SupportsLongNames REG_BINARY 01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Microsoft Exchange Data Source Driver

When you install the Microsoft® Exchange Data Source driver, the Setup program writes a set of default values to the Microsoft Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly; use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Microsoft Exchange Data Source driver.

Microsoft Exchange Data Source Initialization Settings

The Jet\4.0\Engines\Exchange folder includes initialization settings for the msexch40.dll driver, used for external access to Microsoft Outlook and Microsoft Exchange folders. The only entry in this folder is the following:

```
win32=<path>\msexch40.dll
```

The Microsoft Jet database engine uses this setting to indicate the location of msexch40.dll. The full path is determined at the time of installation.

The results of using the Outlook ISAM format and of using the Exchange client ISAM format are similar. The only difference is that the two different clients use different names for the same columns. The two ISAM formats have been created so that Microsoft Jet can return the column names in the particular style that the user desires.

Microsoft Outlook Client ISAM Formats

The Jet\4.0\ISAM Formats\Exchange 9.0 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Exchange
ImportFilter	REG_SZ	Outlook()

CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	3
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
SupportsLongNames	REG_BINARY	01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

Microsoft Exchange Client ISAM Formats

The Jet\4.0\ISAM Formats\Exchange 4.0 folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Exchange
ImportFilter	REG_SZ	Exchange()
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	3
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
SupportsLongNames	REG_BINARY	01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

Customizing the Schema.ini File for Outlook and Exchange Data

The Schema.ini file is used by the Outlook and Exchange ISAM in much the same way that it is used by the Text ISAM. Schema.ini contains the specifics of a data source: how the data is formatted, and the names of columns that should

be accessed.

It is not necessary to modify the Schema.ini file before data can be read, imported, or exported for Outlook and Exchange. Many of the settings inside the Schema.ini file for Outlook and Exchange are specific to internal tags that MAPI requires. You should not attempt to modify those tag values.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Microsoft Jet 4.0 Database Engine Driver

When you install the Microsoft® Jet version 4.0 Engine database driver, the Setup program writes a set of default values to the Microsoft Windows® Registry in the Engines and ISAM Formats subkeys. You must use the Registry Editor to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Microsoft Jet Engine database driver.

Microsoft Jet Engine Initialization Settings

The Jet\4.0\Engines folder includes initialization settings for the msjet40.dll database engine, used for access to Microsoft Access databases. Typical initialization settings for the entries in this folder are shown in the following example.

```
SystemDB = <path>\System.mdb
```

```
CompactBYPkey = 1
```

```
PrevFormatCompactWithUNICODECompression=1
```

The Microsoft Jet database engine uses the following entries.

Entry	Description
SystemDB	Specifies the full path and file name of the workgroup information file. The default is the appropriate path followed by the file name System.mdb. Specifies that when you compact tables they are copied in primary-key order, if a primary key exists on the

CompactByPKey

table. If no primary key exists on a table, the tables are copied in base-table order.

A value of 0 indicates that tables should be compacted in base-table order; a non-zero value indicates that tables should be compacted in primary-key order, if a primary key exists. The default value is non-zero.

Note This setting only applies to compacting databases created with the Microsoft Jet database engine version 3.0 or later; when you compact databases created with the Microsoft Jet database engine version 2.x, the data is always copied in the order of the base table.

Microsoft Jet 4.0 databases use the Unicode character set to store textual data. Compressing the Unicode data can significantly improve the performance of the database because of the reduced number of page read/write operations that are needed afterwards.

This key determines if databases created by the

PrevFormatCompactWithUNICODECompression Microsoft Jet database

engine version 3.x or earlier should be created with compressed Unicode or uncompressed Unicode.

Note This setting does not apply to compacting Microsoft Jet 4.0 databases. Microsoft Jet 4.0 databases will default to keep the compression settings with which they were created.

The Jet\4.0\Engines\Jet 4.0 folder includes initialization settings for the msjet40.dll database engine, used for access to Microsoft Access databases. Typical initialization settings for the entries in this folder are shown in the following example.

FlushTransactionTimeout=500

LockDelay=100

LockRetry=20

MaxBufferSize= 0

MaxLocksPerFile= 9500

PageTimeout=5000

Threads=3

UserCommitSync=Yes

ImplicitCommitSync=No

ExclusiveAsyncDelay=2000

SharedAsyncDelay=0

RecycleLVs=0

PagesLockedToTableLock=0

The Microsoft Jet database engine uses the following entries.

Entry	Description
PageTimeout	<p>The length of time between the time when data that is not read-locked is placed in an internal cache and the time when it is invalidated, expressed in milliseconds. The default is 5000 milliseconds or 5 seconds.</p>
FlushTransactionTimeout	<p>This entry disables both the ExclusiveAsyncDelay and SharedAsyncDelay registry entries. To enable those entries, a value of zero must be entered. FlushTransactionTimeout changes the Microsoft Jet database engine's method for doing asynchronous writes to a database file. Previously, the Microsoft Jet database engine would use either the ExclusiveAsyncDelay or SharedAsyncDelay to determine how long it would wait before forcing asynchronous writes. FlushTransactionTimeout changes that behavior by having a value that will start asynchronous writes only after the specified amount of time has expired and no pages have been added to the cache. The only exception to this is if the cache exceeds the MaxBufferSize, at which point the cache will start asynchronous writing regardless of whether or not the time has expired. Microsoft Jet 3.5 database engine will wait 500 milliseconds of non-activity or until the cache size is exceeded before starting asynchronous writes.</p>
LockDelay	<p>This setting works in conjunction with the LockRetry setting in that it causes each LockRetry to wait 100 milliseconds before issuing another lock request. The LockDelay setting was added to prevent bursting that would occur with certain networking operating systems.</p> <p>This setting prevents transactions in Microsoft Jet from exceeding the specified value. If the locks in a</p>

MaxLocksPerFile

transaction attempt to exceed this value, then the transaction is split into two or more parts and partially committed. This setting was added to prevent Netware 3.1 server crashes when the specified Netware lock limit was exceeded, and to improve performance with both Netware and NT.

LockRetry

The number of times to repeat attempts to access a locked page before returning a lock conflict message. The default is 20.

This setting, when enabled, will cause Microsoft Jet to recycle long value (LV) pages (Memo, Long Binary [OLE object], and Binary data types).

Microsoft Jet 3.0 would not recycle those types of pages until the last user closed the database. If the RecycleLVs setting is enabled, Microsoft Jet 3.5 will start to recycle most LV pages when the database is expanded (that is, when groups of pages are added).

RecycleLVs

Note By enabling this feature, users will notice a performance degradation when manipulating long value data types. Microsoft Access 97 automatically enables and disables this feature when manipulating modules, forms, and reports, thus eliminating the need to turn it on when modifying those objects. The default value is 0.

The size of the database engine internal cache, measured in kilobytes (K). MaxBufferSize must be an integer value greater than or equal to 512. The default is based on the following formula:

$$((\text{TotalRAM in MB} - 12 \text{ MB}) / 4) + 512 \text{ KB}$$

MaxBufferSize

For example, on a system with 32 MB of RAM, the default buffer size is $((32 \text{ MB} - 12 \text{ MB}) / 4) + 512 \text{ KB}$ or 5632 KB. To set the value to the default, set the registry key to

MaxBufferSize=

Threads	The number of background threads available to the Microsoft Jet database engine. The default is 3.
UserCommitSync	Specifies whether the system waits for a commit to finish. A value of Yes instructs the system to wait; a value of No instructs the system to perform the commit asynchronously. The default is Yes.
ImplicitCommitSync	Specifies whether the system waits for a commit to finish. A value of No instructs the system to proceed without waiting for the commit to finish; a value of Yes instructs the system to wait for the commit to finish. The default is No.
ExclusiveAsyncDelay	Specifies the length of time, in milliseconds, to defer an asynchronous flush of an exclusive database. The default value is 2000, or 2 seconds.
SharedAsyncDelay	Specifies the length of time, in milliseconds, to defer an asynchronous flush of a shared database. The default value is 0.
	During bulk operations it is often more efficient to lock a whole table, instead of obtaining locks for each individual page of the table as you try to access it.
	This setting specifies the number of pages that Microsoft Jet will allow to be locked in any particular transaction before Jet attempts to escalate to an exclusive table lock
PagesLockedToTableLock	The default value of 0 indicates that Jet will never automatically change from page locking to table locking.
	Note This setting should be used carefully. If a database is needed for multi-user access, then locking a whole table could cause locking conflicts for other users. This would be especially severe if a small number was used for this setting. Even when a larger number was used, such as 25 or 50, the operation for other users might become

unpredictable.

Microsoft Jet Engine Replication Settings

The Jet\4.0\Transporter key includes initialization settings for Jet Replication synchronizations via the Jet Synchronizer. These settings determine the order that specific transports will be attempted by Synchronizer synchs. The valid values are 0-100. A value of 0 implies that a transport should not be attempted. All transport types with non-zero values will be attempted in ascending key value order until one successfully executes the synchronization or until all attempts fail.

Typical initialization settings for the entries in this folder are shown in the following example.

Priority_FS=1

Priority_Internet=2

Priority_direct=3

The Microsoft Jet database engine uses the following [priority](#) entries.

Entry	Description
Priority_FS	File System (Indirect) Synchronization relies on a series of message exchanges between replicas. The Synchronizer that manages each replica collects changes into message files (*.msg), which are then copied to a shared folder called a dropbox, used by the partner Synchronizer and accessible via the file system.
Priority_Internet	Internet (Indirect) Synchronization relies on a series of message exchanges between replicas. Only the replica managed by the Synchronizer on the Internet Server has a dropbox that is accessible via a HTTP or FTP connection.
Priority_direct	Direct synchronization is the process of exchanging data and design changes between two members of a replica set that are directly connected, either on the same computer or over a

network. The Synchronizer opens both databases.

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Microsoft Jet 3.5 Database Engine Driver

When you install the Microsoft® Jet version 3.5 Engine database driver, the Setup program writes a set of default values to the Microsoft Windows® Registry in the Engines and ISAM Formats subkeys. You must use the Registry Editor to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Microsoft Jet Engine database driver.

Microsoft Jet Engine Initialization Settings

The Jet\4.0\Engines\Jet 3.x folder includes initialization settings for the msrd3x40.dll driver, used for access to Microsoft Access 97 worksheets. Typical initialization settings for the entries in this folder are shown in the following example.

```
win32=<path>\ msrd3x40.dll
```

```
FlushTransactionTimeout=500
```

```
LockDelay=100
```

```
LockRetry=20
```

```
MaxBufferSize= 0
```

```
MaxLocksPerFile= 9500
```

```
PageTimeout=5000
```

```
Threads=3
```

```
UserCommitSync=Yes
```

```
ImplicitCommitSync=No
```

```
ExclusiveAsyncDelay=2000
```

SharedAsyncDelay=0

RecycleLVs=0

SortMemorySource=0

The Microsoft Jet database engine uses the following entries.

Entry	Description
win32	Location of the database engine driver (.dll). The path is determined at the time of installation.
PageTimeout	The length of time between the time when data that is not read-locked is placed in an internal cache and when it is invalidated, expressed in milliseconds. The default is 5000 milliseconds or 5 seconds.
FlushTransactionTimeout	This entry disables both the ExclusiveAsyncDelay and SharedAsyncDelay registry entries. To enable those entries, a value of zero must be entered. FlushTransactionTimeout changes the Microsoft Jet database engine's method for doing asynchronous writes to a database file. Previously, the Microsoft Jet database engine would use either the ExclusiveAsyncDelay or SharedAsyncDelay to determine how long it would wait before forcing asynchronous writes. FlushTransactionTimeout changes that behavior by having a value that will start asynchronous writes only after the specified amount of time has expired and no pages have been added to the cache. The only exception to this is if the cache exceeds the MaxBufferSize, at which point the cache will start asynchronous writing regardless if the time has expired. Microsoft Jet 3.5 database engine will wait 500 milliseconds of non-activity or until the cache size is exceeded before starting asynchronous writes.
LockDelay	This setting works in conjunction with the LockRetry setting in that it causes each LockRetry to wait 100 milliseconds before issuing another lock request. The LockDelay setting was added to prevent "bursting"

that would occur with certain networking operating systems.

MaxLocksPerFile

This setting prevents transactions in Microsoft Jet from exceeding the specified value. If the locks in a transaction attempts to exceed this value, then the transaction is split into two or more parts and partially committed. This setting was added to prevent Netware 3.1 server crashes when the specified Netware lock limit was exceeded and to improve performance with both Netware and NT.

LockRetry

The number of times to repeat attempts to access a locked page before returning a lock conflict message. The default is 20.

RecycleLVs

This setting, when enabled, will cause Microsoft Jet to recycle long value (LV) pages (Memo, Long Binary [OLE object], and Binary data types). Microsoft Jet 3.0 would not recycle those types of pages until the last user closed the database. If the RecycleLVs setting is enabled, Microsoft Jet 3.5 will start to recycle most LV pages when the database is expanded (that is, when groups of pages are added).

Note By enabling this feature, users will notice a performance degradation when manipulating long value data types. Microsoft Access 97 automatically enables and disables this feature when manipulating modules, forms, and reports, thus eliminating the need to turn it on when modifying those objects. The default value is 0.

The size of the database engine internal cache, measured in kilobytes (K). MaxBufferSize must be an integer value greater than or equal to 512. The default is based on the following formula:

$$((\text{TotalRAM in MB} - 12 \text{ MB}) / 4) + 512 \text{ KB}$$

MaxBufferSize

For example, on a system with 32 MB of RAM, the default buffer size is $((32 \text{ MB} - 12 \text{ MB}) / 4) + 512 \text{ KB}$

or 5632 KB. To set the value to the default, set the registry key to

MaxBufferSize=

Threads	The number of background threads available to the Microsoft Jet database engine. The default is 3.
UserCommitSync	Specifies whether the system waits for a commit to finish. A value of Yes instructs the system to wait; a value of No instructs the system to perform the commit asynchronously. The default is Yes.
ImplicitCommitSync	Specifies whether the system waits for a commit to finish. A value of No instructs the system to proceed without waiting for the commit to finish; a value of Yes instructs the system to wait for the commit to finish. The default is No.
ExclusiveAsyncDelay	Specifies the length of time, in milliseconds, to defer an asynchronous flush of an exclusive database. The default value is 2000 or 2 seconds.
SharedAsyncDelay	Specifies the length of time, in milliseconds, to defer an asynchronous flush of a shared database. The default value is 0.
SortMemorySource	Specifies how Microsoft Jet obtains the memory that is used for sort keys. A value of 0 indicates that memory should be taken from the heap. A value of 1 indicates that memory should be taken from global memory using the malloc function call.

Microsoft Jet Engine ISAM Formats

The Jet\4.0\ISAM Formats\Jet 3.x folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Jet 3.x
OneTablePerFile	REG_BINARY	00

IndexDialog REG_BINARY 00
CreateDBOnExport REG_BINARY 00
IsamType REG_DWORD 0

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Microsoft Jet 2.5 Database Engine Driver

When you install the Microsoft® Jet version 2.5 Engine database driver, the Setup program writes a set of default values to the Microsoft Windows® Registry in the Engines and ISAM Formats subkeys. You must use the Registry Editor to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Microsoft Jet Engine database driver.

Microsoft Jet Engine Initialization Settings

The Jet\4.0\Engines\Jet 2.x folder includes initialization settings for the msrd2x40.dll driver, used for access to Microsoft Access 2.0 worksheets. Typical initialization settings for the entries in this folder are shown in the following example.

```
win32=<path>\ msrd2x40.dll
```

```
PageTimeout=5
```

```
LockedPageTimeout=5
```

```
CursorTimeout=5
```

```
LockRetry=20
```

```
CommitLockRetry=20
```

```
MaxBufferSize=512
```

```
ReadAheadPages=16
```

```
IdleFrequency=10
```

```
ForceOsFlush = 0
```

The following entries are used to configure the Microsoft Jet database engine.

Entry	Description
win32	Location of the database engine driver (.dll). The path is determined at the time of installation.
PageTimeout	The length of time between when data that is not read-locked is placed in an internal cache and when it is invalidated, expressed in 100 millisecond units. The default is 5 units (or 0.5 seconds).
LockedPageTimeout	The length of time between when data that is read-locked is placed in an internal cache and when it is invalidated, expressed in 100 millisecond units. The default is 5 units (or 0.5 seconds).
CursorTimeout	The length of time a reference to a page will remain on that page, expressed in 100 millisecond units. The default is 5 units (or 0.5 seconds). This setting applies only to databases created with version 1.x of the Microsoft Jet database engine.
LockRetry	The number of times to repeat attempts to access a locked page before returning a lock conflict message. The default is 20 times; LockRetry is related to CommitLockRetry. The number of times the Microsoft Jet database engine attempts to acquire a lock on data to commit changes to that data. If the Microsoft Jet database engine cannot acquire a commit lock, changes to the data will be unsuccessful.
CommitLockRetry	The number of attempts the Microsoft Jet database engine makes to get a commit lock is directly related to the LockRetry value. For each attempt made to acquire a commit lock, the Microsoft Jet database engine will make as many attempts as specified by the LockRetry value to acquire a lock. For example, if CommitLockRetry is set to 20 and LockRetry is set to 20, the Microsoft Jet database engine will try to acquire a commit lock as many as 20 times; for each of those attempts, the Microsoft Jet database engine will try to acquire a lock as many as 20 times, for a total of 400 attempts.

The default value for CommitLockRetry is 20.

MaxBufferSize	The size of the database engine internal cache, measured in kilobytes (K). MaxBufferSize must be a whole number value between 9 and 4096, inclusive. The default is 512.
ReadAheadPages	The number of pages to read ahead when performing sequential scans. The default is 16.
ForceOSFlush	Any setting other than 0 means a commit or a write will force flushing the OS cache to disk. A setting of 0 (the default setting) means no force flush occurs.
IdleFrequency	The amount of time, in 100 millisecond units, that Microsoft Jet will wait before releasing a read lock. The default is 10 units or one second.

Microsoft Jet Engine ISAM Formats

The Jet\4.0\ISAM Formats\Jet 2.x folder contains the following entries.

Entry name	Value
Engine	Jet 2.x
OneTablePerFile	00
IndexDialog	00
CreateDBOnExport	00
IsamType	0

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Paradox Database Driver

When you install the Paradox database driver, the Setup program writes a set of default values to the Microsoft® Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly (unless necessary); use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Paradox database driver.

The Paradox database driver will work in one of two modes, depending upon whether the Borland Database Engine (BDE) is installed. Paradox data is only updateable with the BDE. Without the BDE, the Paradox data can be Read, Exported, or Linked to read-only.

Paradox Initialization Settings

The Jet\4.0\Engines\Paradox folder includes initialization settings for the Mspdox35.dll driver, used for access to external Paradox data. Typical settings for the entries in this folder are shown in the following example.

```
win32=<path>\ mspbde40.dll
```

```
PageTimeout=600
```

```
CollatingSequence=ASCII
```

```
DataCodePage=OEM
```

```
ParadoxUserName=Kimberly
```

```
ParadoxNetPath=P:\PDOXDB
```

```
ParadoxNetStyle=4.X
```

The Microsoft Jet database engine uses the Paradox folder entries as follows.

Entry

Description

win32 The location of mspbde40.dll. The full path is determined at the time of installation.

PageTimeout The length of time between when data is placed in an internal cache and when it is invalidated. The value is specified in 100 millisecond units. The default is 600 units or 60 seconds.

This setting is only used if the BDE is not present on the machine.

CollatingSequence The collating sequence for all Paradox tables created or opened using the Microsoft Jet database engine. Possible values are ASCII, International, Norwegian-Danish, and Swedish-Finnish. The CollatingSequence entry must match the collating sequence used when the Paradox table was built. The default is ASCII.

This setting is only used if the BDE is not present on the machine.

An indicator of how text pages are stored. Possible settings are:

DataCodePage

- OEM— OemToAnsi and AnsiToOem conversions done.
- ANSI— OemToAnsi and AnsiToOem conversions not done.

The default is OEM.

The name to be displayed by Paradox if a table is locked by the Paradox ISAM and an interactive user accessing the data from Paradox (rather than the ISAM) attempts to place an incompatible lock. This entry is not added by the setup program. You must create this entry.

ParadoxUserName **Note** If you indicate a ParadoxUserName, you must also specify a ParadoxNetPath and a ParadoxNetStyle or you will receive an error when trying to access external Paradox data. Also, if you are accessing a Paradox database in multi-user

mode over the network, you will need to add or modify this registry entry manually.

The full path to the directory containing the PARADOX.NET file (for Paradox 3.x) or the PDOXUSRS.NET file (for Paradox 4.x). This entry is not added by the setup program. You must create this entry. The full ParadoxNetPath (including the drive letter) must be consistent for all users sharing a particular database (directory).

ParadoxNetPath

Note If you indicate a ParadoxNetPath, you must also specify a ParadoxUserName and a ParadoxNetStyle or you will receive an error when trying to access external Paradox data. Also, if you are accessing a Paradox database in multi-user mode over the network, you will need to add or modify this registry entry manually.

The network access style to use when accessing Paradox data. Possible values are:

- 3.x
- 4.x

Note Paradox 3.x users cannot set this to 4.x or the driver will use the wrong locking method. Paradox 5.0 users must use the 4.x ParadoxNetStyle setting to ensure proper locking behavior.

ParadoxNetStyle

This entry is not added by the setup program. You must create this entry. This entry should correspond to whatever version of Paradox the users in the group are using. It must be consistent for all users sharing a particular database (directory). The default is 4.x.

Note If you indicate a ParadoxNetStyle, you must also specify a ParadoxUserName and a ParadoxNetPath or you will receive an error when trying to access external Paradox data.

Paradox ISAM Formats

The Jet\4.0\ISAM Formats\Paradox 3.x folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Paradox
ExportFilter	REG_SZ	Paradox 3 (*.db)
ImportFilter	REG_SZ	Paradox (*.db)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a Paradox 3 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

The Jet\4.0\ISAM Formats\Paradox 4.x folder contains the following entries.

Entry name	Type	Value
------------	------	-------

Engine	REG_SZ	Paradox
ExportFilter	REG_SZ	Paradox 4 (*.db)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextExport	REG_SZ	Export data from the current database into a Paradox 4 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

The Jet\4.0\ISAM Formats\Paradox 5.x folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Paradox
ExportFilter	REG_SZ	Paradox 5 (*.db)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextExport	REG_SZ	Export data from the current database into a Paradox 5 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

The Jet\4.0\ISAM Formats\Paradox 7.x folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Paradox
ExportFilter	REG_SZ	Paradox 7 (*.db)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	0
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextExport	REG_SZ	Export data from the current database into a Paradox 7 file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	00

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Initializing the Text Data Source Driver

The same database driver is used for both Text Data sources and for HTML data sources.

When you install the Text Data Source database driver, the Setup program writes a set of default values to the Microsoft® Windows® Registry in the Engines and ISAM Formats subkeys. You should not modify these settings directly; use the setup program for your application to add, remove, or change these settings. The following sections describe initialization and ISAM Format settings for the Text Data Source database driver.

Text Data Source Initialization Settings

The Jet\4.0\Engines\Text folder includes initialization settings for the Mstext35.dll driver, used for external access to text data files. Typical settings for the entries in this folder are shown in the following example.

```
win32=<path>\ mstext40.dll
```

```
MaxScanRows=25
```

```
FirstRowHasNames=True
```

```
CharacterSet= ANSI
```

```
Format=CSVDelimited
```

```
Extensions= txt,csv,tab,asc
```

```
ExportCurrencySymbols=Yes
```

The Microsoft Jet database engine uses the Text folder entries as follows.

Entry	Description
win32	The location of mstext40.dll. The full path is determined at the time of installation.

MaxScanRows	<p>The number of rows to be scanned when guessing the column types. If set to 0, the entire file will be searched. The default is 25.</p>
FirstRowHasNames	<p>A binary value that indicates whether the first row of the table contains column names. A value of 01 indicates that, during import, column names are taken from the first row. A value of 00 indicates no column names in the first row. The default is 01.</p>
CharacterSet	<p>An indicator of how text pages are stored. Possible settings are:</p> <ul style="list-style-type: none"> • ANSI— The ANSI code page of the machine. AnsiToUnicode and UnicodeToAnsi conversions done. • OEM— The OEM code page of the machine. OemToUnicode and UnicodeToOem conversions done. • Unicode— codepage conversions not done. • <decimal number>— The code page number of a specific character set. Conversions to and from Unicode will be done. <p>The default is ANSI.</p>
Format	<p>Can be any of the following: TabDelimited, CSVDelimited, Delimited (<single character>). The single-character delimiter in the Delimited format can be any single character except a double quotation mark ("). The default is CSVDelimited.</p>
Extensions	<p>The extension of any files to be browsed when looking for text-based data. The default is txt, csv, tab, asc.</p>
ExportCurrencySymbols	<p>A binary value that indicates whether the appropriate currency symbol is included when currency fields are exported. A value of 01 indicates that the symbol is included. A value of 00 indicates that only the numeric data is exported. The default is 01.</p>

Text Data Source ISAM Formats

The Jet\4.0\ISAM Formats\Text folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Text
ExportFilter	REG_SZ	Text Files (*.txt; *.csv; *.tab; *.asc)
ImportFilter	REG_SZ	Text Files (*.txt; *.csv; *.tab; *.asc)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	2
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
ResultTextExport	REG_SZ	Export data from the current database into a text file. This process will overwrite the data if exported to an existing file.
SupportsLongNames	REG_BINARY	01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

HTML Import ISAM Formats

The Jet\4.0\ISAM Formats\HTML Import folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Text
ImportFilter	REG_SZ	HTML Files (*.ht*)
CanLink	REG_BINARY	01
OneTablePerFile	REG_BINARY	00
IsamType	REG_DWORD	2
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00
ResultTextImport	REG_SZ	Import data from the external file into the current database. Changing data in the current database will not change data in the external file.
ResultTextLink	REG_SZ	Create a table in the current database that is linked to the external file. Changing data in the current database will change data in the external file.
SupportsLongNames	REG_BINARY	01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

HTML Export ISAM Formats

The Jet\4.0\ISAM Formats\HTML Export folder contains the following entries.

Entry name	Type	Value
Engine	REG_SZ	Text
ExportFilter	REG_SZ	HTML Files (*.htm)
CanLink	REG_BINARY	00
OneTablePerFile	REG_BINARY	01
IsamType	REG_DWORD	2
IndexDialog	REG_BINARY	00
CreateDBOnExport	REG_BINARY	00

ResultTextExport REG_SZ

Export data from the current database into a text file. This process will overwrite the data if exported to an existing file.

SupportsLongNames REG_BINARY 01

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

Customizing the Schema.ini File for Text and HTML Data

To read, import, or export text and HTML data, you need to create a Schema.ini file in addition to including the Text ISAM information in the .ini file.

Schema.ini contains the specifics of a data source: how the text file is formatted, how it is read at import time, and what the default export format is for files. The following examples show the layout for a fixed-width file, Filename.txt:

```
[Filename.txt]
```

```
ColNameHeader=False
```

```
Format=FixedLength
```

```
FixedFormat= RaggedEdge
```

```
MaxScanRows=25
```

```
CharacterSet=OEM
```

```
Col1=columnname Char Width 24
```

```
Col2=columnname2 Date Width 9
```

```
Col3=columnname7 Float Width 10
```

```
Col4=columnname8 Integer Width 10
```

```
Col5=columnname9 LongChar Width 10
```

Similarly, the format for a delimited file is specified as follows:

```
[Delimit.txt]
```

ColNameHeader=True

Format=Delimited()

MaxScanRows=0

CharacterSet=OEM

Col1=username char width 50

Col2=dateofbirth Date width 9

If you are exporting data into a delimited text file, specify the format for that file as well:

[Export: My Special Export]

ColNameHeader=True

Format=TabDelimited

MaxScanRows=25

CharacterSet=OEM

DateTimeFormat=mm.dd.yy.hh.mm.ss

CurrencySymbol=Dm

CurrencyPosFormat=0

CurrencyDigits=2

CurrencyNegFormat=0

CurrencyThousandSymbol=,

CurrencyDecimalSymbol=.

DecimalSymbol=,

NumberDigits=2

NumberLeadingZeros=0

TextDelimiter=""

The My Special Export example refers to a specific export option; you can specify any variation of export options at connect time. This last example also corresponds to a data source name (DSN) that can be optionally passed at connect time. All three format sections can be included in the same .ini file.

The Microsoft Jet database engine uses the Schema.ini entries as follows.

Entry	Description
ColNameHeader	Can be set to either True (indicating that the first record of data specifies the column names) or False .
Format	Can be set to one of the following values: TabDelimited, CSVDelimited, Delimited (<single character>), or FixedLength. The delimiter specified for the Delimited file format can be any single character except a double quotation mark ("). Only used when the Format is FixedLength, this can be set to one of the following values: RaggedEdge or TrueFixedLength. RaggedEdge allows rows to be terminated with a Carriage Return character.
FixedFormat	TrueFixedLength requires each row to be an exact number of characters, and any Carriage Return characters not at a row boundary are assumed to be embedded in a field. If this setting is not present, the default value is RaggedEdge.
MaxScanRows	Indicates the number of rows to be scanned when guessing the column data types. If this is set to 0, the entire file is searched.
CharacterSet	Can be set to OEM, ANSI, UNICODE, or the decimal number of a valid code page, and indicates the character set of the source file. Can be set to a format string indicating dates and times. This entry should be specified if all date/time

DateTimeFormat

fields in the import/export are handled with the same format. All of the Microsoft Jet database engine formats except AM and PM are supported. In the absence of a format string, the Windows Control Panel short date picture and time options are used.

CurrencySymbol

Indicates the currency symbol to be used for currency values in the text file. Examples include the dollar sign (\$) and Dm. If this entry is absent, the default value in the Windows Control Panel is used.

Can be set to any of the following values:

Currency symbol prefix with no separation (\$1)

Currency symbol suffix with no separation (1\$)

CurrencyPosFormat

Currency symbol prefix with one character separation (\$ 1)

Currency symbol suffix with one character separation (1 \$)

If this entry is absent, the default value in the Windows Control Panel is used.

CurrencyDigits

Specifies the number of digits used for the fractional part of a currency amount. If this entry is absent, the default value in the Windows Control Panel is used.

Can be one of the following values:

(\$1)

-\$1

\$-1

\$1-

(1\$)

-1\$

	1-\$
	1\$-
	-1 \$
CurrencyNegFormat	-\$ 1
	1 \$-
	\$ 1-
	\$ -1
	1- \$
	(\$ 1)
	(1 \$)

The dollar sign is shown for purposes of this example, but it should be replaced with the appropriate CurrencySymbol value in the actual program. If this entry is absent, the default value in the Windows Control Panel is used.

CurrencyThousandSymbol Indicates the single-character symbol to be used for separating currency values by thousands in the text file. If this entry is absent, the default value in the Windows Control Panel is used.

CurrencyDecimalSymbol Can be set to any single character that is used to separate the whole from the fractional part of a currency amount. If this entry is absent, the default value in the Windows Control Panel is used.

DecimalSymbol Can be set to any single character that is used to separate the integer from the fractional part of a number. If this entry is absent, the default value in the Windows Control Panel is used.

NumberDigits Indicates the number of decimal digits in the fractional portion of a number. If this entry is absent, the default value in the Windows Control Panel is used.

NumberLeadingZeros Specifies whether a decimal value less than 1 and greater than -1 should contain leading zeros; this value can either be False (no leading zeros) or True.

Lists the columns in the text file to be read. The format of this entry should be:

Coln=columnName type [Width #]

columnName: Column names with embedded spaces should be enclosed in quotation marks.

type: Can be Bit, Byte, Short, Long, Decimal, Currency, Single, Double, DateTime. Binary, OLE, Text, or Memo.

In addition, the following ODBC Text Driver types are supported:

Col1, Col2, ...

Char (same as Text)

Float (same as Double)

Integer (same as Short)

LongChar (same as Memo)

Date *date format*

In the case of a Memo type an additional format marker [Attribute [Hyperlink](#)] can be used to specify columns that should be active URLs in Microsoft Access.

In the case of a Decimal type the additional format markers [Scale #] Precision #] should be used.

Can be set to any single character that is used to delimit strings that contain any of the other special characters.

TextDelimiter

E.g. “abc”, ”xyz,pqr”, ”hij”

If this entry is not present the default delimiter is a double quote. If this entry is the string “none” then no characters will be treated as delimiters.

Note When you change Schema.ini file settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Configuring the Microsoft Jet Database Engine for ODBC Access

The following sections describe Microsoft® Windows® Registry settings for the Microsoft Jet database engine for connection to an ODBC database.

Initialization Settings for Microsoft Jet-connected ODBC Databases

The `\HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\4.0\Engines\ODBC` folder contains initialization settings for the Microsoft Jet database engine.

Note Typical settings for the entries in the `Jet\4.0\Engines\ODBC` folder are shown in the following example.

```
LoginTimeout=20
```

```
QueryTimeout=60
```

```
ConnectionTimeout=600
```

```
AsyncRetryInterval=500
```

```
AttachCaseSensitive=0
```

```
AttachableObjects='TABLE', 'VIEW', 'SYSTEM TABLE', 'ALIAS', 'SYNONYM'
```

```
SnapshotOnly=0
```

```
TraceSQLMode=0
```

```
TraceODBCAPI=0
```

```
DisableAsync=1
```

```
TryJetAuth=1
```

```
PreparedInsert=0
```

```
PreparedUpdate=0
```

FastRequery=0

FatBlastRows=-1

FatBlastTimeout=3

ODBCISAMAttach=0

The Microsoft Jet database engine uses the ODBC entries as follows.

Entry	Description
LoginTimeout	The number of seconds a login attempt can continue before timing out. The default is 20 (values are of type REG_DWORD).
QueryTimeout	The number of seconds a query can run (total processing time) before timing out. If DisableAsync=0 (the default), then QueryTimeout is the number of seconds to wait for a response from the server between polls for query completion. The default is 60 (values are of type REG_DWORD).
ConnectionTimeout	The number of seconds a cached connection can remain idle before timing out. The default is 600 (values are of type REG_DWORD).
AsyncRetryInterval	The number of milliseconds between polls to determine if the server is done processing a query. This entry is used for asynchronous processing only. The default is 500 (values are of type REG_DWORD).
AttachCaseSensitive	An indicator of whether to match table names exactly when linking. Values are 0 (link the first table matching the specified name, regardless of case) and 1 (link a table only if the name matches exactly). The default is 0 (values are of type REG_DWORD).
AttachableObjects	A list of server object types to which linking will be allowed. The default is: 'TABLE', 'VIEW', 'SYSTEM TABLE', 'ALIAS', 'SYNONYM' (values are of type REG_SZ).
SnapshotOnly	An indicator of whether Recordset objects are forced to be of snapshot type. Values are 0 (allow dynasets) and 1 (force snapshots only). The default is 0 (values are of type

	REG_DWORD).
TraceSQLMode	An indicator of whether the Microsoft Jet database engine will trace SQL statements sent to an ODBC data source in SLOUT.txt. Values are 0 (no) and 1 (yes). The default is 0 (values are of type REG_DWORD). This entry is interchangeable with SQLTraceMode.
TraceODBCAPI	An indicator of whether to trace ODBC API calls in ODBCAPI.txt. Values are 0 (no) and 1 (yes). The default is 0 (values are of type REG_DWORD).
DisableAsync	An indicator of whether to force synchronous query execution. Values are 0 (use asynchronous query execution if possible) and 1 (force synchronous query execution). The default is 1 (values are of type REG_DWORD).
TryJetAuth	An indicator of whether to try using the Microsoft Access user name and password to log in to the server before prompting. Values are 0 (no) and 1 (yes). The default is 1 (values are of type REG_DWORD).
PreparedInsert	An indicator of whether to use a prepared INSERT statement that inserts data in all columns. Values are 0 (use a custom INSERT statement that inserts only non- Null values) and 1 (use a prepared INSERT statement). The default is 0 (values are of type REG_DWORD). Using prepared INSERT statements can cause Nulls to overwrite server defaults and can cause triggers to execute on columns that were not inserted explicitly.
PreparedUpdate	An indicator of whether to use a prepared UPDATE statement that updates data in all columns. Values are 0 (use a custom UPDATE statement that sets only columns that have changed) and 1 (use a prepared UPDATE statement). The default is 0 (values are of type REG_DWORD). Using prepared UPDATE statements can cause triggers to execute on unchanged columns.
	An indicator of whether to use a prepared SELECT

FastRequery statement for parameterized queries. Values are 0 (no) and 1 (yes). The default is 0 (values are of type REG_DWORD).

FatBlastRows

FatBlastTimeout

ODBCISAMAttach

Note When you change Windows Registry settings, you must exit and then restart the database engine for the new settings to take effect.

See Also

[Customizing Windows Registry Settings for Microsoft Jet](#)

Using the Connection Properties in the Microsoft OLE DB Provider for Jet

To customize the Microsoft® Windows® Registry settings, you can use the connection properties in the Microsoft OLE DB Provider for Jet. This is accomplished by referencing a property in the connection object and changing its value. For example, assuming that your connection object is called `ADOConnection`, the following would yield the same results as going through ADO:

```
ADOConnection.Properties("Jet OLEDB:Max Locks Per File") = 20000
```

The property names are different than the DAO constants and the registry settings. The property names are as follows:

Jet OLEDB:Max Locks Per File

Jet OLEDB:Implicit Commit Sync

Jet OLEDB:Flush Transaction Timeout

Jet OLEDB:Lock Delay

Jet OLEDB:Max Buffer Size

Jet OLEDB:User Commit Sync

Jet OLEDB:Lock Retry

Jet OLEDB:Exclusive Async Delay

Jet OLEDB:Shared Async Delay

Jet OLEDB:Page Timeout

Jet OLEDB:Recycle Long-Valued Pages

See Also

[Using Regedit.exe to Overwrite the Default Settings](#) [Using the SetOption Method from DAO](#)
[Creating a Portion in Your Application's Registry Tree to Manage the Settings](#)

Using the SetOption Method from DAO

DAO version 3.6 provides a new way to modify default settings. Microsoft® Windows® Registry settings can now be modified at run time with the **SetOption** method. To customize the Windows Registry settings, you can use the **SetOption** method from DAO. With this option, your application obtains the maximum flexibility and control. This approach allows you to create applications that are easier to maintain and that are tuned for maximum performance.

The syntax for doing this is `dbEngine.SetOption, constant, NewValueSetting`. For example, the following syntax, `dbEngine.SetOption dbMaxLocksPerFile, 20000`, would allow Microsoft Jet to track 20,000 locks at one time. The names of defined constants are the same as the registry name with `db` added as a prefix.

This is the recommended way to fine tune registry settings for your application. This method is the most flexible approach and provides you with the most control over how the registry is changed. With The **SetOption** method you can specify new settings for any of the following default settings:

- PageTimeout key
- SharedAsyncDelay key
- ExclusiveAsyncDelay key
- LockRetry key
- UserCommitSync key
- ImplicitCommitSync key
- MaxBufferSize key
- MaxLocksPerFile key
- LockDelay key
- RecycleLVs
- FlushTransactionTimeout key

See Also

[Using Regedit.exe to Overwrite the Default Settings](#)

[Using the Connection Properties in the Microsoft OLE DB Provider for Jet](#)

[Creating a Portion in Your Application's Registry Tree to Manage the Settings](#)

[Show All](#)

About data access pages

A data access page is a special type of Web page designed for viewing and working with data from the Internet or an intranet— data that is stored in a [Microsoft Access database](#) or a [Microsoft SQL Server database](#). The data access page may also include data from other sources, such as Microsoft Excel.

 [A page in Page view or Microsoft Internet Explorer 5.01 with Service Pack 2 \(SP2\) or later](#)

Using a data access page is similar to using a form: You can view, enter, edit, and delete data in a database. However, you can also use a page outside a [Microsoft Access database](#), so users can update or view data over the Internet or an intranet.

The following illustration shows how the Product data access page, which is used for entering and updating product information, looks in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.



1 Enter product information in the text boxes, drop-down list box, and check box.

2 Use the record navigation toolbar to navigate, add, delete, save, sort, and filter records, and to get Help.

[A page in Design view](#)

The following illustration shows the Products data access page in [Design view](#).



- 1 Title in the body of the page
- 2 Group header for the Products group level
- 3 Record navigation section for the Products group level

[Parts of a data access page](#)

The body The body is the basic design surface of a data access page. On a page that supports data entry, you can use it to display informational text, controls bound to data, and sections.

Sections You use sections to display text, data from a database, and toolbars.

Two types of [sections](#) are typically used on pages that support data entry: group header and record navigation sections. A page can also have footer and caption sections.

- **Group header and footer** Used to display data and calculate values.
- **Record navigation** Used to display the [record navigation control](#) for the group level. A record navigation section for a group appears after the group header section. You can't place [bound controls](#) in a record navigation section.

- **Caption** Used to display captions for text boxes and other controls. It appears immediately before the group header. You can't place bound controls in a caption section.

Each group level in a data access page has a [record source](#). The name of the record source is displayed on the [section bar](#) for each section used for a group level.

[Designing different types of data access pages](#)

You design data access pages in [Design view](#) in Microsoft Access. The page is a separate file that is stored outside Access; however, when you create the file, Access automatically adds a shortcut to the file in the Database window. Designing a data access page is similar to designing forms and reports—you use a [field list](#), the [toolbox](#), [controls](#), and so on. However, there are some significant differences in the way that you design and interact with data access pages as opposed to forms and reports. How you design the page depends on what it will be used for:

- **Interactive reporting** This type of data access page is often used to consolidate and group information that is stored in the database, and then publish summaries of the data. For example, a page might publish the sales performance for each region in which you do business. Using [expand indicators](#), you can go from a general summary of the information, such as a list of all the regions and their combined sales total, to specific details on individual sales within each region. The data access page might provide toolbar buttons for sorting and [filtering](#) the data, as well as for adding, editing, and deleting the data in some or all [group levels](#).
- **Data analysis** This type of data access page may include a [PivotTable list](#), similar to a Microsoft Excel [PivotTable report](#), that lets you reorganize the data to analyze it in different ways. The page might contain a chart that you can use to analyze trends, detect patterns, and compare data in your database. Or it might contain a spreadsheet, in which you can enter and edit data and use formulas to calculate as you do in Excel.

[Using data access pages in Internet Explorer](#)

A data access page is connected directly to a database. When users display the data access page in Internet Explorer, they are viewing their own copy of the

page. That means any filtering, sorting, and other changes they make to the way the data is displayed— including changes they make within a [PivotTable list](#) or [spreadsheet](#)— affect only their copy of the data access page. However, changes that they make to the data itself— such as modifying values, and adding or deleting data— are stored in the underlying database, and therefore are available to everyone viewing the data access page.

Users get Help on how to work with the page in Internet Explorer by clicking the **Help**  button on the record navigation toolbar. The Help file that appears is automatically included with any data access page that's published with a record navigation toolbar. If you delete the record navigation toolbar or if you disable its **Help** button, you should provide instructions for using the page to those who will use the page.

Note To view and work with the data access page on the Internet or an intranet, users need Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

[Using data access pages in Microsoft Access](#)

You can also work with a data access page in [Page view](#) in Access. Data access pages can supplement the forms and reports that you use in your database application. When deciding whether to design a data access page, form, or report, consider the tasks that you want to perform.

You get Help on how to work with the page in Page view by clicking the **Help**  button on the **Page View** toolbar to display Access Help. Clicking the **Help** button on a record navigation toolbar on the page displays Help for working with the page in Microsoft Internet Explorer. As mentioned previously, this Help file is automatically included with any data access page that's published with a record navigation toolbar. You can delete the **Help** button on pages that are used only within Access, or you can modify the button to provide custom Help.

[Where a data access page gets its data](#)

[Types of data sources for a data access page](#)

A data access page gets its data from a [Microsoft Access database](#), or a [Microsoft SQL Server database](#) version 6.5 or later. To design a page using data

from one of these databases, the page must be connected to the database. If you have already opened an Access database or an [Access project](#) that's connected to a SQL Server database, the data access page you create automatically connects to the current database and stores that path in the **ConnectionString** property of the data access page. When a user browses to the page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later or displays the page in [Page view](#), it displays current data from the underlying database by using the path defined in the **ConnectionString** property. If you design the data access page while the database is on a local drive, Access will use the local path, which means the data won't be accessible to others. For this reason, it's important that you move or copy the database to a network location that's accessible to users of your page. Once the database is on the network share, open the database using the [UNC](#) address. If you move or copy the database after you've already designed the page, you'll have to update the path in the **ConnectionString** property to point to the new location.

Instead of having to update the **ConnectionString** property of each page in the database separately, you can choose to create a connection file. A connection file stores the connection information for a data access page, and can be shared between multiple data access pages. When you open a data access page that uses a connection file, the page reads the connection file and connects to the appropriate database. After creating the connection file, if you move or copy the database, all you have to do is edit the connection information in the connection file.

 [Types of data sources for Microsoft Office Web Components on a data access page](#)

Although the data access page gets its data from a [Microsoft Access database](#) or [Microsoft SQL Server database](#), [Microsoft Office Web Component](#) controls on the page can display data from either of these databases or from other sources. For example, the page may contain a [PivotTable list](#), [spreadsheet](#), or [chart](#) with data from a Microsoft Excel worksheet or a database other than Access or SQL Server. Depending on the data source, the data in these controls may be a snapshot of the original data or it may be live data displayed through a separate connection that the control has with its data source.

To create an Office Web Component with data from a source other than the current database, you might have to use a program other than Access to first

publish the data to a Web page. For example, to create a PivotTable list that contains data from an Excel worksheet, you have to publish the data from Excel. Then, in Access, you can open the Web page and turn it into a data access page by adding controls that are bound to an Access or SQL Server database. You can further customize the page by adding labels, pictures, or other features to enhance the page's appearance. From within Access, you can create a PivotTable list that uses data from an external source other than Excel.

Security Note Data access pages aren't actually stored in the Access file; they are stored as HTML files either in the local file system, in a folder on a network share, or on an [HTTP](#) server. For this reason, Microsoft Access has no control over the security of data access page files. To help protect the data access page, you must employ security measures for its link and HTML file by using the file system security of the computer where you have stored these files. To help protect the data accessed by the page, you must either employ security measures for the database that the page is connected to or configure Microsoft Internet Explorer security settings to prevent unauthorized access.

[Show All](#)

Troubleshoot data access pages

[Sorting data](#)

[I get the message "The DefaultSort property is invalid."](#)

In the **DefaultSort** property box, type the name of the field or fields you want to sort on. Separate multiple field names with a comma. If you want to sort in ascending or descending order, you can follow each name with a space and the keyword ASC or DESC, respectively. For example, type **OrderDate DESC, OrderID ASC**. If you don't specify the order, Microsoft Access sorts in ascending order.

If the control you want to sort on is bound to an [expression](#), you must type the alias of the control as displayed in its **ControlSource** property. For example, if the **ControlSource** property of the control you want to sort on is set to **GroupOfExtendedPrice: ExtendedPrice**, type **GroupOfExtendedPrice** in the **DefaultSort** property box.

[Calculating data](#)

[I get the message "The expression is not valid."](#)

You attempted to type an [expression](#) that includes the name of a field that is not in the [data definition](#) of the page. Any field that you reference in an expression on a data access page must be in that page's data definition. To add a field to the page's data definition, drag the field from the [field list](#) to the data access page. If you don't want to show the [values](#) in the field on the page, you can hide or delete the control bound to the field. However, even if you delete the control, a field referenced in an expression remains part of the page's data definition.

[I get the message "Undefined function in an expression."](#)

You've bound a control on a data access page to a [calculated field](#) that uses a function that isn't supported on data access pages.

[Functions supported on a data access page](#)

A [control](#) in a [data access page](#) or a form that is open in [PivotTable view](#) or [PivotChart view](#) can contain an [expression](#) that uses a Microsoft Visual Basic for Applications (VBA) function. However, the expression will generate an error unless it appears on the following list.

Note This limitation doesn't apply to expressions in tables, queries, views, stored procedures, or user-defined functions that you have opened in PivotTable or PivotChart view.

[A - C](#)

ABS ARRAY
ASC ASCB
ASCW ATN
CBOOL CBYTE
CCUR CDATE
CDBL CHOOSE
CHR CHR\$
CHRB CHRB\$
CHRW CHRW\$
CINT CLNG
COS CSNG
CSTR CVAR
CVDATE CVERR

[D - H](#)

DATE DATE\$
DATEADD DATEDIFF
DATEPART DATESERIAL
DATEVALUE DAY
DDDB ERROR
ERROR\$ EXP

FIX	FORMAT
FORMAT\$	FV
HEX	HEX\$
HOUR	

[I - L](#)

IIF	IMESTATUS
INSTR	INT
IPMT	IRR
ISDATE	ISEMPTY
ISERROR	ISNULL
ISNUMERIC	ISOBJECT
ICASE	ICASE\$
LEFT	LEFT\$
LEFTB	LEFTB\$
LEN	LENB
LOG	LTRIM
LTRIM\$	

[M - Q](#)

MID	MID\$
MIDB	MIDB\$
MINUTE	MIRR
MONTH	NOW
NPER	NPV
OCT	OCT\$
PARTITION	PMT
PPMT	PV
QBCOLOR	

[R - S](#)

RATE RGB
RIGHT RIGHT\$
RIGHTB RIGHTB\$
RND ROUND
RTRIM RTRIM\$
SECOND SGN
SIN SLN
SPACE SPACE\$
SQR STR
STR\$ STRCOMP
STRCONV STRING
STRING\$ SWITCH
SYD

[T - Z](#)

TAN TIME
TIME\$ TIMER
TIMESERIAL TIMEVALUE
TRIM TRIM\$
TYPENAME UCASE
UCASE\$ VAL
VARTYPE WEEKDAY
YEAR

[Designing a page](#)

[I can't select multiple controls](#)

To select multiple controls, you must have Microsoft Internet Explorer 5.5 or later installed on your computer.

[I have the same version of Microsoft Access as the rest of my team, but my design environment seems different from that of the rest of my team.](#)

Check the version of Microsoft Internet Explorer that is installed on your computer. If you don't have Internet Explorer version 5.5 or later, you will not be able to select and work with multiple controls, or see some properties and toolbars.

[I get the message "Invalid argument."](#)

- In the regional settings of Microsoft Windows Control Panel, your location is set to a region that uses a character other than a period (.) as a decimal separator. The settings for properties in data access pages require that the decimal separator be a period for all languages. Either change the region or set the decimal symbol to a period in Windows Control Panel regional settings.
- The [value](#) you typed in the property sheet does not match the [data type](#) of the property. For example, the **Width** property of a section can be set to any number, but if you type a string such as **Narrow**, you will get the "Invalid argument" message.

[Some wizards are not available](#)

- The Command Button Wizard is not available on a stand-alone data access page— a page that you open without having a database open. This wizard also is not available if the page isn't bound to a table or [query](#).
- The List Box Wizard and the Combo Box Wizard aren't available on stand-alone data access pages. The wizards also are not available when the page is connected to a database other than the one that's currently open.

[I get the message "The setting you entered isn't valid for this property."](#)

If you are entering a [value](#) in a property that has a drop-down list, select a value from the list. For information about valid values for properties that don't have drop-down lists, press F1 in the property box.

If you are entering an [expression](#) in the **ControlSource** property box, make sure:

- You precede the expression with an equal sign (=) or an alias for the expression. For example, both of the following are valid settings:

=Quantity*UnitPrice

ExtendedPrice: Quantity*UnitPrice

- The data access page is bound to a table, [query](#), or other [record source](#) before you attempt to enter the expression. You can't use expressions on a data access page that isn't bound to data.
- The **ID** property of a [calculated control](#) is set to a unique name. Don't use the name of one of the fields you used in the expression or the ID of another control on the data access page.

[Working with files and links](#)

[I don't want to keep my supporting data access page files in a folder.](#)

When you save a data access page in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default stored in a separate folder. If you do not want to use a separate folder, do the following:

1. Open the data access page in Design view.
2. On the **Tools** menu, click **Options**, and then click the **General** tab.
3. Click **Web Options**, and then click the **Files** tab.
4. Under **File names and locations**, clear the **Organize supporting files in a folder** check box.

[I moved my data access page to another location and now some of the links are broken.](#)

When you save your file as a data access page in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. If you move or copy your data access page to another location, you must also move the supporting folder so that you maintain all links to your page.

[I can't automatically create a backup copy of my data access page.](#)

When you save a data access page in [Design view](#), a backup copy is not automatically created. To save a copy of a data access page, click **Save As** on the **File** menu. In the **Save Data Access Page** <object name> **To** box, type a new name for the data access page, and then click **OK**.

[The name of a supporting folder for a data access page is in another language.](#)

When you save your data access page in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. By default, the name of the supporting folder is the name of the data access page plus an underscore (_), a period (.), or a hyphen (-) and the word "files." The word "files" will appear in the language of the version of Microsoft Office that was used to save the file as a data access page. For example, suppose you use the Dutch language version of Office to save a file called Page1 as a data access page. The default name of the supporting folder would be **Page1_bestanden**.

Note If you save your data access page with a short file name (a maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the data access page without the word "files."

[I can't locate the HTML file corresponding to a data access page.](#)

The [HTML](#) file associated with the data access page has been renamed, moved, or deleted.

Although a data access page is a [database object](#), most of the information about a page is stored in a corresponding HTML file. You specify the name and location of this HTML file the first time you create a data access page. Microsoft Access keeps a shortcut in the [Database window](#) to the corresponding HTML file's location in your file system. You can see the HTML file's location in the Database window by resting the pointer on the name of the page.

If the HTML file has been renamed or moved, you can restore the HTML file's original name, move the HTML file back to its original location, or change the shortcut maintained by Access to the new name or location. (Click **Locate** in the message box that appears when Access cannot locate the HTML file.)

Notes

- It's possible to have two or more pages in the same or different [Microsoft Access databases](#) or [Microsoft Access projects](#) pointing to the same HTML file. For example, you might have copied or imported the page. Although you might want to do this, it's a good idea to modify the page in [Design view](#) in only one Access database or Access project at a time to avoid problems with sharing and to avoid possibly losing design changes.
- If the HTML file is deleted, you cannot recover the page unless you can retrieve the HTML file from a backup copy.

[Using graphics](#)

[I see a red X, a blank image, or no image at all on my data access page.](#)

You might have the wrong browser settings Although all browsers support the same language ([HTML](#)), there are some differences among browsers. Some support different versions of HTML and different [HTML extensions](#). Custom options can be set in browsers, such as the default text and background colors, and whether or not graphics are displayed. Consider testing your data access page in different browsers— some of which are available for downloading on the Web — and with different monitor settings, to see if all of your page elements appear.

You might have used the wrong picture format If you link a graphic to a data access page, and the graphic format isn't supported by your browser, your graphics won't be visible. Make sure that the linked picture is in a graphic format that is compatible with your browser. The [JPEG](#) and [GIF](#) graphic formats are compatible with most browsers.

[The graphics and text on my data access page look different on another computer.](#)

If your data access page looks different from what you expect, check the browser settings. Although all Web browsers support the same language ([HTML](#)), there are some differences among browsers. Some support different versions of HTML and different [HTML extensions](#). Custom options can be set in browsers, such as the default text and background colors, and whether or not graphics are

displayed. Consider testing your page in different browsers— some of which are available for downloading on the Web— and with different monitor settings, to see whether all of your page elements appear.

[I chose the **Active Graphics** option when I applied the theme, but none of the graphics are animated.](#)

Theme graphics are animated only when you view the data access page in a Web browser, not in Microsoft Access.

[Specifying a language](#)

[My data access page is displaying the wrong characters for a language.](#)

If a data access page is [encoded](#) for a different language, Microsoft Access tries to determine the language. If Access displays the wrong characters for that language when you open the page in a browser, you can select the language that you think the page is encoded in. Some languages have more than one encoding, so try each encoding until you can read the text.

1. Open the data access page in [Design view](#).
2. On the **Tools** menu, click **Options**, and then click the **General** tab.
3. Click **Web Options**, and then click the **Encoding** tab.
4. Under **Save this document as**, select the language that you think the page is encoded in.

[The name of a supporting folder for a data access page is in another language.](#)

When you save your data access page in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. By default, the name of the supporting folder is the name of the data access page plus an underscore (_), a period (.), or a hyphen (-) and the word "files." The word "files" will appear in the language of the version of Microsoft Office that was used to save the file as a data access page. For example, suppose you use the Dutch language version of Office to save a file

called Page1 as a data access page. The default name of the supporting folder would be **Page1_bestanden**.

Note If you save your data access page with a short file name (a maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the data access page without the word "files."

[Open and viewing](#)

[I'm having problems viewing a data access page.](#)

- You might not have the correct Web browser.

To view or interact with a data access page, you must have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later installed on your computer. A page can also contain additional controls, including spreadsheets, PivotTable lists, and charts.

- Another user might have exclusive access to the [Microsoft Access database](#) or [Microsoft SQL Server database](#) that the page is connected to.

This situation can occur when a database administrator is maintaining an Access database or SQL Server database and needs to temporarily prevent users from accessing the database, or when someone is designing objects in a shared Access database. Wait a few minutes and try to view the page again, or contact your database administrator.

- The page, or a PivotTable list on the page, is connected to an unsupported OLE DB data source.

An Access database or Microsoft SQL Server database is the only OLE DB data source you can connect to from a page. Either you or the author of the page needs to change the OLE DB connection to the page. To do so, in [Design view](#), right-click the database icon in the [field list](#), and then click **Connection**.

[I want to open an HTML file in Access, but it opens in another Office program.](#)

In the **Open** dialog box (**File** menu), when you select an [Hypertext Markup Language](#) (HTML) file (also called a Web page) and click **Open** or double-click the file, the file opens in the program that created the file. For example, if you attempt to open an HTML format file that was created in Microsoft Word, the file opens in Word— not in Microsoft Access as a data access page.

To open an HTML file that was created in another Office program as a data access page in Access, right-click the file in the **Open** dialog box, point to **Open with**, and then click **Microsoft Access** on the shortcut menu.

Note If the file type that you are trying to open hasn't previously been associated with another program, instead of pointing to the **Open with** dialog, click the **Open with** dialog, and select **Microsoft Access** from the list of programs.

[Working in Page view](#)

[Buttons are missing from the record navigation toolbar.](#)

The designer of the data access page removed the buttons for that page or group.

[The fields on the data access page are blank.](#)

Possible reasons include:

- The data access page you are displaying might be a data-entry form. You can enter your data in the blank fields.
- The data access page contains a drop-down list that determines which records to display. From the drop-down list, select a group whose records you want to see on the page.

[I'm having problems editing data in a field in Page view.](#)

There are several kinds of fields that you can't edit in [Page view](#). The following fields might display values just as regular fields do, but if you try to enter data into them, nothing happens:

- **AutoNumber fields** If a field has an AutoNumber data type in a

[Microsoft Access database](#) or a numeric data type with the **Identity** property set in a [Microsoft Access project](#), Access automatically assigns a number to this field for each record you add. This type of field is often used as a record ID number or [primary key](#). Access automatically fills in the field's value for you.

- **Calculated fields** In an Access database, a [calculated field](#) displays values that Access calculates. They are usually based on other fields in your tables, but calculated fields are not stored in tables.
- **Read-only or disabled controls** If a control on a form has the **ReadOnly** property set to **True** or the **Disabled** property set to **True**, it is locked or disabled for data entry.
- **Fields in a locked record** If you use an Access database in a multiuser environment and a record is locked by another user, you can't edit the data in the record until the other user is finished editing the record.
- **Fields in a read-only form or a locked database** If the form you're using has the **AllowEdits** property set to **No** or if the underlying data is read-only or locked, you can't edit the data.
- **Fields in a read-only group level on a data access page** A read-only [group level](#) on a page is intended only to display data.
- **A read-only PivotTable list** A PivotTable list might be intended only to display data, not to edit data.
- **Fields in a table that doesn't support edits** If the underlying table does not have a primary key, or if the **UniqueTable** property of the corresponding group header section is not set to the name of the table, you can't edit data.

You might be having problems entering dates in a field with the [Date/Time data type](#). If you are entering dates by using the slash mark (/) format, don't enter a trailing slash mark. For example, 1/3 is valid, but 1/3/ is not valid.

[I sent a data access page as an e-mail message but the recipient can't read it.](#)

To view a data access page sent as an e-mail message, the mail recipient must read the e-mail message by using one of the following:

- Microsoft Outlook 2000 or later
- Microsoft Outlook Express 5.0 or later

- The combination of Microsoft Outlook 98 or later and Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later
- The combination of Outlook Express 5.0 or later and Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later

If you are using a corporate intranet, you can publish your pages from a Web site located in a trusted security zone. This will help alleviate problems caused by the recipient's Internet Explorer security settings and thus improve performance of your pages. For more information about security levels and security zones, see Internet Explorer Help. For any specific page security requirements at your site, see your system administrator.

[I get a message when I attempt to enter data in Page view.](#)

When you try to move out of a field after changing data, Microsoft Access validates the data, making sure that the [value](#) you entered is allowed in that field. If the value isn't allowed, Access alerts you. In order to move off the field, you must either change the data to an acceptable value or undo your changes.

A value might not be allowed in a field for the following reasons:

- The value isn't compatible with the field's data type. If you enter a value that doesn't match the data type, Access attempts to convert the entry to the correct data type. However, some data can't be converted to other data types. For example, you can't store a name in a field with the Number data type in a [Microsoft Access database](#) or the smallint data type in a [Microsoft Access project](#).
- The data breaks a rule set in the **ValidationRule** property for the field.
- Data is required in the field, and you've tried to leave it blank.
- The validation macro specified for the **BeforeUpdate** property canceled the update.

Access also validates data when you move out of a record. Before it saves a record, Access makes sure that your changes don't break any [validation rules](#) set in the **ValidationRule** property for the record. Additionally, if the table you are editing has a [primary key](#) or [unique index](#), Access makes sure that the record you are editing doesn't have the same value for the fields as another record in the table. If the record can't be saved, you must either make any necessary

corrections, or cancel your changes by clicking **Undo**  or pressing ESC.

Note The data access page might contain a PivotTable Component, Spreadsheet Component, or Chart Component, which has its own troubleshooting information. For more information, click **Help**  on the toolbar for the [PivotTable list](#), [spreadsheet](#), or [chart](#), and then search for "troubleshooting."

[Show All](#)

About securing a data access page

Data access pages are [Hypertext Markup Language \(HTML\)](#) pages that contain references to the data in a Microsoft Access file. However, the data access pages aren't actually stored in the Access file; they are stored as .htm files in the local file system, in a folder on a network share, or on an [HTTP](#) server. For this reason, Access has no control over the security of data access page files. To help protect the data access page, you must employ security measures for its link and HTML file. To help protect the data accessed by the page, you must either employ security measures for the database that the page is connected to or configure Microsoft Internet Explorer security settings.

[Securing a data access page link and its corresponding HTML file](#)

A data access page is the combination of a link stored in the [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp) file and a corresponding HTML file located in your computer. To prevent a page's link and its corresponding HTML file from being renamed, deleted, or changed, use the file security of the computer where you have stored these files. For example, on your computer, you can make the Access database or Access project file read-only. On the Web server where the page's corresponding HTML file is located (after you have published the page), make the file and the folder where the HTML file is located read-only. For more information about file and folder security, see Microsoft Windows Help.

[Helping protect and control database access from a data access page](#)

Consider the following when you want to help protect access to the Access database or Microsoft SQL Server database that the page is connected to:

- **Help protect the database** As with any application, you need to help prevent unauthorized users from accessing the database, and help control the level of access granted once authorized users log on.
 - For Access databases with user-level security, if you want to enable user-level security through a page, you need to make sure that the

connection information for the page specifies the correct [workgroup information file](#) and that this file is in a network share accessible to all users. Modify the connection to the page in the **Data Link Properties** dialog box by clicking the **All** tab, double-clicking **Jet OLEDB:System database**, and then specifying the [UNC](#) path to the correct workgroup information file in the **Property Value** box.

If you set a database password on an Access database that is connected to a page, users will be prompted for the password the first time they view the page.

- For an SQL Server database, if you want to use Microsoft Windows integrated security to connect from a page, modify the connection to the page in the **Data Link Properties** dialog box by clicking **Connection**, and then clicking **Use Windows Integrated security**.
- **Don't save the user name and password with the page** If the user name and password to the database are saved with the page, any user can log on to the database. If you want the page to prompt the user to log on to the database, modify the connection to the page in the **Data Link Properties** dialog box by clicking the **Connection** tab, and then clearing the **Allow saving of password** check box.
- **Return records based on a user name** You might want to create a page that doesn't require entering a user name and password, but can automatically detect who the user is and display records based on that user. For example, you could create a page that allows users to change their personnel file. The general steps for doing this are:
 - Create a page in an Access project connected to an SQL Server database that employs Windows integrated security technology.
 - Add a column to the employee table that has each employee's Windows log on user name.
 - Create a view that uses the `suser_sname()` function to detect the user currently logged on, and make that view the page's [record source](#). For more information about the `suser_sname()` function, see Microsoft SQL Server Help.

[Preventing unauthorized access by a malicious script](#)

A data access page uses the Microsoft Office Data Source Control (MSODSC) to connect to its data source. When a user opens a page, the MSODSC on the page may use the user name of that user to log on to the database. A malicious page author could try to exploit this and attempt to access databases on servers that are accessible to the user but not to the page author. The mode of data access used by the MSODSC determines whether a page meets industry-standard security levels, or if cross-domain data access is possible from the page. You can configure the MSODSC to use two modes of data access: two-tier or three-tier data access.

Two-tier data access is a direct connection to a data source in a traditional client/server fashion—the client (first tier) makes a direct connection to the database server (second tier). To perform two-tier data access, the MSODSC makes a direct connection to its data source through an OLE DB provider that is registered on the user's computer. For an Access database, two-tier data access uses the local computer's Microsoft Jet 4.0 OLE DB provider to connect to an .mdb file located on a network share. For an SQL Server database, two-tier data access uses the local computer's Microsoft OLE DB provider for SQL Server to connect to a database server available on your local network.

Three-tier data access refers to using a third, remote, component between the client and database components. Three-tier data access is typically used to access data across the Internet. For three-tier data access that uses the MSODSC, this third component is the [ActiveX Data Objects \(ADO\) Remote Data Service \(RDS\) component](#) (also called Data Factory) running on Microsoft Internet Information Services (IIS).

Any page that uses two-tier data access is considered by Microsoft Internet Explorer to be making a cross-domain access attempt. Depending on the security settings in Internet Explorer, when a user opens a page by using two-tier data access, one of three things occurs: the page is automatically disabled, the user is asked whether to allow data access, or the page is automatically enabled. Depending on future security patches and upgrades, these options may be different for your system. You should check the Web or ask your system administrator for additional information about new security patches or upgrades.

Internet Explorer security settings also define different security zones. If a page is on a Web server in a trusted zone, the cross-domain attempt can be enabled automatically. In a controlled environment, such as a corporate intranet, your

pages will perform better if you use two-tier data access and make sure that they are published from a server located in a trusted security zone. This is the simplest way to help protect against unauthorized access from malicious scripts. For more information about security settings, see Microsoft Internet Explorer Help.

A page that uses three-tier data access to connect to a database is considered to meet industry-standard security levels regardless of what Internet Explorer security zone it is published from. This means that Internet Explorer will not warn the user about cross-domain data access attempts when three-tier pages are opened if authentication settings have been left to their default configuration.

[Show All](#)

Differences between forms, reports, and data access pages

Each Microsoft Access [database object](#) is designed for a particular purpose. In the following table, **Yes** identifies the objects that are best at performing a particular task, **Possible** identifies objects that can perform the task, though less optimally, and **No** identifies objects that don't perform the task at all.

Task/Purpose	Form	Report	Data access page
Enter, edit, and interact with data within a Microsoft Access database or Microsoft Access project	Yes	No	Yes
Enter, edit, and interact with live data via the Internet or an intranet outside an Access database or Access project; users must have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later	No	No	Yes
Print data for distribution	Possible	Yes	Possible
Distribute data by e-mail	No	No	Yes (live or static data)

Data access pages have the following advantages over printed reports:

- Pages bound to data display current data because they are connected to a database.
- Pages are interactive. Users can filter, sort, and view just the records that they want.
- Pages can be distributed electronically by using e-mail. Recipients will see current data each time they open the message.

[Show All](#)

Differences in page support between Internet Explorer 5.01 with Service Pack 2 (SP2) and 5.5 or later

You will notice some differences in data access page [Design view](#) depending on the version of Microsoft Internet Explorer that you have on your computer. Although Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) is adequate to design and use a page, Internet Explorer 5.5 or later provides additional design and browse capabilities.

 [When designing a data access page in Microsoft Access](#)

Task	Supported in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2)	Supported in Internet Explorer 5.5 or later
Select multiple controls in page Design view	No	Yes. You can select multiple controls, and move, resize, align, or format them.
Preserve a control's unit of measurement during a resizing operation	No	Yes
Live resizing of controls (the edges of the control adjust immediately based on mouse pointer movement)	No. The control doesn't resize until after the mouse pointer is released.	Yes
View settings of properties supported by Internet Explorer 5.5 or later in data access page Design view	No	Yes. Some Internet Explorer 5.5 or later properties are displayed in the page, section, and control property sheets.

Both Internet Explorer 5.01 with Service Pack 2 (SP2) and Internet Explorer 5.5 or later support aligning controls on a data access page in Design view, but the steps and the user interface are different.

[When viewing a data access page in Internet Explorer](#)

Task	Supported in Internet Explorer 5.01 with Service Pack 2 (SP2)	Supported in Internet Explorer 5.5 or later
Print preview a data access page	No	Yes

You might notice some differences in the performance of a page, depending on the version of Internet Explorer. A page used with Internet Explorer 5.5 or later might be faster than when used with Internet Explorer 5.01 with Service Pack 2 (SP2).

[Show All](#)

System requirements for sending and viewing a data access page

The following are system requirements for sending and viewing a [data access page](#).

System requirements for sending a data access page

To send a data access page through e-mail you need one of the following:

- Microsoft Outlook 2000 (or later version)
- Outlook Express version 5.0 (or later version)
- Microsoft Exchange

System requirements for viewing a data access page sent through e-mail

To view a data access page sent as an e-mail message, recipients need one of the following:

- Microsoft Outlook 2000 (or later version)
- Outlook Express version 5.0 (or later version)
- The combination of Microsoft Outlook 98 (or later version) and Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later
- The combination of Outlook Express 98 (or later version) and Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later

Notes

- Sending a data access page through e-mail might not work across electronic mail gateways because of differences in security settings between domains or security restrictions on e-mail attachments. You should check with your system administrator if you are having problems receiving or viewing data access pages sent as e-mail.
- It is recommended that you send a data access page as an attachment in an e-mail message if the recipient is using Microsoft Office 97 or earlier.

[Show All](#)

Publish Web pages

After you create a Web page, you can [publish](#) the page to Web Folders or a Web server.

[Publish a data access page to Web Folders or any Web server](#)

Important A data access page is connected to either a Microsoft Access or Microsoft SQL Server database. For the page to work properly, the database must be available to users of the page.

[To publish a data access page file by saving it to Web Folders](#)

- When you first create the page, save it to a [Web folder](#) underneath Web Folders. If you create the page in a location other than a Web folder underneath Web Folders, you can move the corresponding HTML file and related files and folders by using the **Save As** command on the **File** menu in Access.

Notes

- Access automatically saves related files such as graphics, style sheets, linked files, and the folders that contain these files when you save any Web page to Web Folders.
- You can only use the Web Folders shortcut or enter a Uniform Resource Locator ([URL](#)) in the **File name** box of the **Open** dialog box when you edit a Web page that already exists or create a data access page in page Design view.

[To publish a data access page to a Web server](#)

- Copy the HTML file corresponding to the page and any other related files and folders in Windows Explorer to a folder underneath the [root directory](#) of the Web server. The default root directory is **\Webshare\Wwwroot** for Personal Web Server and **\Inetpub\Wwwroot** for Microsoft Internet

Information Server.

[Publish server-generated HTML files to a Web server](#)

To use server-generated HTML files (either [ASP](#) or [IDC/HTX](#)) on a Web server, you need to do the following.

1. Install the appropriate software on the computer that will be the Web server and that will process the server-generated HTML files.
2. Create a folder under the [root directory](#) to contain the ASP or IDC/HTX files. The default root directory is **\Webshare\Wwwroot** for Personal Web Server and **\Inetpub\Wwwroot** for Microsoft Internet Information Server.
3. Copy the ASP or IDC/HTX files to the folder. Copy related files such as graphics, linked files, and any folders that may contain these files to the folder, or make sure the related files can be located by the Web server software.
4. Either copy the Microsoft Access database or SQL Server database to the folder, or define its network location in the [ODBC data source](#) definition in step 6.
5. Define the ODBC data source as a [System DSN](#) on the Web server. Make sure that the data source name is the same name as the one you entered in the appropriate **Output Options** dialog box when you output the ASP or IDC/HTX files.
6. If desired, define database security for the ODBC data source, but ensure that users can access the ODBC data source, by doing one of the following:
 - For a [Microsoft Access database](#) (.mdb), create a [user-level security](#) username and password to allow users to access the Access database from the Web page. If you do not create a user-level security username or password, the default username is **Admin**, and no password is used.

Warning This will allow anyone with default Administrator privileges to access your database from the Web page. It is highly recommended that Administrator accounts have strong passwords assigned to them. Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh%et5 Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

- For a [Microsoft Access project](#) (.adp), in the **Username** and **Password** boxes, create a database username and password to allow users to access the Microsoft SQL Server database from the Web page. If you do not create a username or password, the default username is **Sa**, and no password is used.

Warning This will allow anyone with default Administrator privileges to access your database from the Web page. It is highly recommended that Administrator accounts have strong passwords assigned to them. Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh%et5 Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

The username and password must match the username and password that you enter in the **User to Connect As** and **Password for User** boxes of the **Microsoft Active Server Pages Output Options** or **HTX/IDC Output Options** dialog boxes that appear when you output the ASP or IDC/HTX files.

Note If you need to export multiple datasheets, you can automate the operation by creating a macro or creating a Visual Basic for Applications program. This is particularly useful when you frequently export datasheets to test and build a Web site.

[Publish static HTML files to Web Folders or any Web server](#)

1. Do one of the following:
 - To publish a static HTML file to Web Folders, copy the static HTML file(s) to a Web folder under Web Folders by using Windows Explorer.
 - To publish a static HTML file to a Web server, copy the static HTML file to a folder under the [root directory](#) of the Web server by using Windows Explorer. The default root directory is **\Webshare\Wwwroot** for Personal Web Server and **\Inetpub\Wwwroot** for Microsoft Internet Information Server.
2. Copy related files such as graphics, style sheets, linked files, and any folders that may contain these files to the folder, or make sure the related

files can be located by the Web server.

Note If you need to export multiple tables, queries, forms, and reports, you can automate the operation by creating a macro or creating a Visual Basic for Applications program. This is particularly useful when you frequently export tables, queries, forms, and reports to test and build a Web site.

[Show All](#)

Default names for data access page supporting folders

When you save your [data access page](#) in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. By default, the name of the supporting folder is the name of the data access page plus an underscore (_), a period (.), or a hyphen (-) and the word "files." The word "files" will appear in the language of the version of Microsoft Office 2000 or later that was used to save the file as a data access page. For example, suppose you use the Dutch language version of Office 2000 or later to save a file called Page1 as a data access page. The default name of the supporting folder would be **Page1_bestanden**.

Note If you save your data access page with a short file name (maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the data access page without the word "files."

The following table shows the translation for "files" in each language version of Office 2000 or later.

Office 2000 language Default name for Web page supporting folder

Arabic	".files"
Basque	"_fitxategiak"
Brazilian	"_arquivos"
Bulgarian	".files"
Catalan	"_fitxers"
Chinese (Simplified)	".files"
Chinese (Traditional)	".files"
Croatian	"_datoteke"
Czech	"_soubory"
Danish	"-filer"
Dutch	"_bestanden"

English	"_files"
Estonian	"_failid"
Finnish	"_tiedostot"
French	"_fichiers"
German	"-Dateien"
Greek	".files"
Hebrew	".files"
Hungarian	"_elemei"
Italian	"-file"
Japanese	".files"
Korean	".files"
Latvian	"_fails"
Lithuanian	"_bylos"
Norwegian	"-filer"
Polish	"_pliki"
Portuguese	"_ficheiros"
Romanian	".files"
Russian	".files"
Serbian (Cyrillic)	".files"
Serbian (Latin)	"_fajlovi"
Slovakian	".files"
Slovenian	"_datoteke"
Spanish	"_archivos"
Swedish	"-filer"
Thai	".files"
Turkish	"_dosyalar"
Ukrainian	".files"
Vietnamese	".files"

[Show All](#)

About sending a data access page in e-mail

When you send a [data access page](#) in e-mail, there are important connection and security considerations you need to be aware of to ensure more reliable results.

[Share the database](#)

A data access page is connected to one of two types of data sources: a [Microsoft Access database](#) or a [Microsoft SQL Server database](#). For the page to work properly when sent via e-mail, the source database must be on a shared server or shared computer in order for users to view and interact with the page.

[Use UNC paths](#)

Before you create the page, open the [Microsoft Access database](#) by using a [universal naming convention \(UNC\)](#) path in the **File name** box on the **Open** dialog box (**File** menu, **Open** command). This ensures that the data source of the page is defined with a UNC path instead of with the drive letter of a mapped network drive in Microsoft Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for the page to be located.

When you create the page, you should save it using a UNC path in the **File name** box of the **Save As Data Access Page** dialog box. Saving your page with a UNC path helps protect you when you send a pointer to the page by e-mail.

[Store the database and the page on the same server](#)

To further help protect your pages, put the [Microsoft Access database](#) or [Microsoft SQL Server database](#) on the same [Web server](#) as the data access page.

Note You can't copy an Access database to or create an Access database in a folder under Network Places.

[Publish from a trusted intranet security zone](#)

Depending on the Microsoft Internet Explorer security level setting on the recipient's computer, one of three things occurs when the recipient opens an e-mail message containing a data access page: the page is automatically disabled, the recipient is asked whether to allow data access, or the page is automatically enabled.

If you are using a corporate intranet, you can publish your pages from a Web site located in a trusted security zone. This will help alleviate problems caused by the recipient's Internet Explorer security settings and thus improve performance of your pages. For more information about security levels and security zones, see Internet Explorer Help. For any specific page security requirements at your site, see your system administrator.

[Send a pointer instead of a copy of the HTML source code](#)

There are three ways Microsoft Access can send a data access page in an e-mail message.

- In the body of the message, as a copy of the HTML source code of the page
- In the body of the message, as a pointer to the HTML file associated with the page
- As an attachment to the message, including a copy of the HTML file associated with the page

In the e-mail message, a copy of the HTML source code of a page looks very different than a pointer to the page. The copy will contain all of the HTML tags and objects needed to render the page. The pointer will contain some HTML tags and the absolute path— a [universal resource locator \(URL\)](#), or a [universal naming convention \(UNC\)](#) path— to the page.

It's best to avoid sending a copy of the HTML source code of a page because Microsoft Outlook won't let a user connect to data across domains unless the recipient or system administrator has changed the default Internet Explorer security zone settings. Sending a pointer to the page assumes that the page and the data source that the page connects to are located on a server or public share. This reduces the chance for cross-domain issues within Microsoft Outlook.

Sending a pointer instead of a copy of the page can also be a security precaution that reduces the risk of malicious users intercepting a copy of the HTML source code of a page that has been sent to a user, and then revising it to covertly gain access to a corporate database.

The best way to send a data access page is to send a pointer to the page in the body of the e-mail message. To send a pointer, you can do one of the following:

- Save the page to a folder under Network Places or on a Web server, so the page's location is a URL.
- Save the page to a shared computer using a UNC path.

In either case, Access uses the <IFRAME> HTML tag to send a pointer to the page. The pointer is either the URL or the UNC path to the page.

When you create your e-mail message, Access provides confirmation dialog boxes to help you make the right choice.

To help maintain security when sending a page, first send a pointer to the page to yourself and then open the e-mail message. In the message window, right-click the page, and then click **View Source**. If the HTML text looks similar to the following code, then you are sending a pointer to the page (and not a copy of the HTML source code):

```
<HTML><BODY STYLE="OVERFLOW:HIDDEN" BOTTOMMARGIN=0  
LEFTMARGIN=0 TOPMARGIN=0 RIGHTMARGIN=0>  
<IFRAME FRAMEBORDER=0 HEIGHT=100% WIDTH=100%  
SRC="http://NWweb/Northwind pages/View Products.htm"></IFRAME>  
</BODY><HTML>
```

[Show All](#)

Strategies for creating data access pages

You can use many different features to create effective data access pages, depending on what you want to accomplish. The following information helps you identify the right features and layout to use for each type of page.

[Analyze data](#)

Use a [PivotTable list](#) to analyze data because it allows you to reorganize the data in different ways. You can bind a PivotTable list to data in the database or use data from a Microsoft Excel worksheet. You can use the PivotTable list as the only [control](#) on a data access page or use it with other controls.

- When using a PivotTable list on an ungrouped page, place the PivotTable list and other controls either in the [body](#) or in a [section](#) of the page.
- Decide if you want to allow users to add, edit, or delete data, and then set the [group level](#) properties accordingly.
- When using a PivotTable list on a grouped data access page:
 - Place the PivotTable list and [bound controls](#) in a section.
 - Make sure the PivotTable list is in the lowest group level.
 - If the PivotTable list is the only control in a section, delete or hide the [navigation section](#) for the group level that the PivotTable list is in.

[Enter and edit data](#)

You can create a data access page on which users can enter, edit, and delete data in the database. Key points to keep in mind when creating a page for data entry:

- Use individual controls, such as [text boxes](#), [list boxes](#), [drop-down list boxes](#), [option groups](#), [option buttons](#), and [check boxes](#).
- Place controls either in the [body](#) or in a [section](#) of the page. You can delete a section if you don't use it.
- Use the [Spreadsheet Component](#) to create a spreadsheet to perform one or more calculations on the fields in a record. You can display the calculated

values in the spreadsheet or hide the spreadsheet and display the calculated values in [bound span controls](#).

- Determine how users can use the data access page by customizing the record navigation toolbar.

[Make projections](#)

You can use the [Chart Component](#) to create a chart to analyze trends, show patterns, and make comparisons on the data in your database, or use the [Spreadsheet Component](#), in which you can enter and edit data and use formulas to perform some of the same calculations you can do in Microsoft Excel.

- When using a spreadsheet on an ungrouped data access page, place the spreadsheet and other controls either in the [body](#) or in a [section](#) of the page.
- When using a spreadsheet on a grouped data access page, place the spreadsheet and other controls bound to fields in the database in a section. You can use a spreadsheet in any [group level](#).

[Review data](#)

Data access pages give you a way to interact with large amounts of data in a selective way. By expanding and collapsing groups of records, users can focus only on the data that they want to see. Keep the following points in mind when you create a grouped page:

- Place [bound controls](#) in a [section](#).
- In the lowest [group level](#), use individual controls or a [PivotTable list](#) to display detail records.
- Remove unneeded buttons from the record navigation toolbar.
- Set the properties of the group levels to make them read-only.
- To make loading the data access page in Microsoft Internet Explorer faster:
 - Use [bound span controls](#) instead of text boxes.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

- Set the **ExpandedByDefault** property to **False** for all group levels.
- Set the **DataPageSize** property of the group levels to a low rather than a high number. The lower the number, the faster the records are displayed.

Notes

- Microsoft Access 2000 doesn't allow users to add, edit, or delete data in a grouped page. When you open an Access 2000 page in Access 2002 or later, group levels are read-only by default. You can change the group level properties to enable additions, edits, and deletions. However, you can't edit the grouping field.
- When a group level includes fields from more than one table or query, only fields from a single table can support data entry. The **UniqueTable** property of the header section has the name of the object that is updatable. You can set this property to an underlying table or— if the page includes fields from a query— to a table underlying the query.

[Show All](#)

Create a data access page

To create a [data access page](#), or to open a page in [Design view](#), [Page view](#), or in Microsoft Internet Explorer, you must install Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

If you create a data access page when a database is open, Microsoft Access creates a shortcut to the page and displays the shortcut in the [Database window](#). If you create a page without opening a database, Access will create a stand-alone page.

[Create a data access page in the current database](#)

[Create a data access page that contains all fields from a single record source \(AutoPage\)](#)

Use the AutoPage command to create a data access page that contains all records and fields (except fields that store pictures) in the underlying table, query, or view.

1. Under **Objects** in the Database window, click **Pages** .
2. On the Database window toolbar, click **New**.
3. In the **New Data Access Page** dialog box, click **AutoPage: Columnar**.
4. Click the table, query, or view that contains the data on which you want to base your page.
5. Click **OK**.

In the finished data access page, each field will appear on a separate line with a label to its left. If the resulting page doesn't look the way you want, you can modify the page in Design view.

Note If you have set a default [theme](#), Microsoft Access applies it to the page.

[Create a data access page that contains fields from multiple tables and queries \(Page Wizard\)](#)

The wizard asks you detailed questions about the [record sources](#), fields, layout, and format you want, and then creates a page based on your answers.

1. Under **Objects** in the Database window, click **Pages** .
2. On the Database window toolbar, click **New**.
3. In the **New Data Access Page** dialog box, click **Page Wizard**.
4. Click the name of the table, query, or view that includes the data on which you want to base your data access page.

Note You don't need to do this step now—you can specify the record sources for the page in the wizard.

5. Click **OK**.
6. Follow the directions in the wizard dialog box.

If the resulting page doesn't look the way you want, you can modify the page in [Design view](#).

Notes

- If you want to include fields from multiple tables and queries in your page, don't click **Next** or **Finish** after selecting the fields from the first table or query in the Page Wizard. Repeat the steps to select a table or query, and then choose fields that you want to include in the page, until you have selected all the required fields.
- If you include fields from multiple tables or queries but don't create [group levels](#), fields from one of the underlying tables can support updates in Page view or in Microsoft Internet Explorer. The wizard will list the record sources of the page and prompt you to specify the updatable record source.

[Create a data access page in Design view](#)

1. Under **Objects** in the Database window, click **Pages** .
2. On the Database window toolbar, click **New**.
3. In the **New Data Access Page** dialog box, click **Design View**.
4. Click the name of the table, query, or view that includes the data on which you want to base your data access page.

Note If you want to create a blank page, don't select a record source in this dialog box.

5. Click **OK**.

Microsoft Access displays the data access page in Design view, where you can modify the page. To add data to the page, drag fields from the field list to the page.

[Make an existing Web page into a data access page](#)

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Data Access Page** dialog box, click **Existing Web page**.
4. Click **OK**.
5. In the **Locate Web Page** dialog box, find the Web page or HTML file that you want to open.

Note You can find a Web page by clicking **Search the Web**  in the **Locate Web Page** dialog box. Save a copy of the page by using the **Save As** command on the **File** menu in Microsoft Internet Explorer, and then start this procedure again to open the page in [Design view](#).

6. Click **Open**.

Microsoft Access creates a shortcut to the HTML file in the Database window and displays the page in Design view, where you can modify the page.

Note If you have specified a default connection file for the database in the **Pages** tab of the **Options** dialog box, Access will automatically set the **ConnectionFile** and **ConnectionString** properties for all new pages.

[Create a stand-alone data access page](#)

1. If a database is open in Microsoft Access, close it.
2. On the **File** menu, click **New**.
3. In the **New File** dialog box, click **Blank Data Access Page**.
4. In the **Select Data Source** dialog box, do one of the following:

[Connect the page to a Microsoft Access database](#)

Use the following procedure if you want to create a page and bind it to an Access database, without creating a shortcut in the database.

1. In the **Files of Type** list, select **Access databases**.
2. Double-click the database file that you want to bind the page to. If you don't see the database that you want to use, navigate to the folder that has the file, and then double-click the file name.
3. Click **OK** twice.

Access will create a blank page and open it in Design view. The page's **ConnectionString** property will be set to connect the page to the database you selected. To add data to the page, drag fields from the field list to the page.

[Connect the page to a SQL Server database or an OLE DB data source by creating a connection file](#)

1. Click **New Source**.
2. In the **Data Connection Wizard** dialog box, select the data source that you want to connect to.

3. Follow the directions in the wizard dialog box.
4. In the **Select Data Source** dialog box, click the connection file that you created in the previous step.
5. Do one of the following:

- To use the connection information without creating a link between the page and the connection file, click **Open**.

Access will create a blank page and open it in Design view. The page's **ConnectionString** property will be set based on the contents of the connection file you created. To add data to the page, drag fields from the field list to the page.

- To create a link between the page and the connection file, click the arrow on the **Open** button, and then click **Use Connection File**.

Access will create a blank page and open it in Design view. The page's **ConnectionString** property will be set based on the contents of the connection file, and the **ConnectionFile** property will be set to the connection file that you created. To add data to the page, drag fields from the field list to the page.

[Connect the page to a data source using an existing connection file](#)

1. In the **Files of Type** list, select **Office Database Connections**.
2. Click the connection file that has the connection information that you want to use for the page. If you don't see the connection file that you want to use, navigate to the folder that has the file, and then click the file name.
3. Do one of the following:
 - To use the connection information without creating a link between the page and the connection file, click **Open**.

Access will create a blank page and open it in Design view. The page's **ConnectionString** property will be set based on the

contents of the connection file you selected. To add data to the page, drag fields from the field list to the page.

- To create a link between the page and the connection file, click **Use Connection File**.

Access will create a blank page and open it in Design view. The page's **ConnectionString** property will be set based on the contents of the connection file, and the **ConnectionFile** property will be set to the connection file that you selected. To add data to the page, drag fields from the field list to the page.

[Show All](#)

Convert an existing Web page into a data access page

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. In the **New Data Access Page** dialog box, click **Existing Web page**.
4. Click **OK**.
5. In the **Locate Web Page** dialog box, find the Web page or HTML file that you want to open.

Note You can find a Web page by clicking **Search the Web**  in the **Locate Web Page** dialog box. Save a copy of the page by using the **Save As** command on the **File** menu in Microsoft Internet Explorer, and then start this procedure again to open the page in [Design view](#).

6. Click **Open**.

Microsoft Access creates a shortcut to the HTML file in the Database window and displays the page in Design view, where you can modify the page.

[Show All](#)

Convert a previous-version data access page

A page created using Microsoft Access 2000 must be converted before it can be used with a later version.

Note You must install Microsoft Office 2000 Web Components that is included with Microsoft Office 2000 or Microsoft Office XP on your computer before you attempt to convert the page.

- In Access 2002 or later, open the page in [Design view](#).

Access will create a backup copy of the page (*pagefilename.bak.htm*) at the same location as the original data access page file.

Notes

- If you want to revert to the previous version of the page, delete the converted file, rename the backup copy, and then connect the page to the database.
- If the data access page includes any Microsoft Office Web Components (a [PivotTable list](#), [chart](#), or [spreadsheet](#)), only users with a valid Microsoft Office license will be able to use those components.

[Show All](#)

About using connection files

A connection file stores information about a connection to a data source (such as an OLE DB data source) and the data associated with the connection. In Microsoft Access, you can use a connection file to bind one or more [data access pages](#) to a data source. When you open a page, Access reads the connection file that is linked to the page, and based on the contents of the connection file, connects the page to the appropriate data source.

Security Note Under some circumstances, a user can view the information used to connect to an external data source. This may represent a security risk. For more information, see [About making connections to external data sources more secure](#).

A connection file uses one of the following file formats:

- Office Data Connection (.odc)

A file format that uses HTML and XML to store the connection information. You can view or edit the contents of the file in any text editor.

- Universal Data Link (.udl)

A standard file format provided by Microsoft Data Links to create file-persistent OLE DB data source object definitions.

[Benefits of using a connection file](#)

Using a connection file simplifies the task of deploying related data access pages. Pages that connect to a common data source can share a single connection file. If you move or copy the data source, instead of updating the **ConnectionString** property of each dependent page, you only need to edit the connection information in the connection file to make the pages point to the right location or database.

[Linking a page to a connection file](#)

When creating a data access page, you can choose whether you want to link the page to a connection file or simply use a connection file without creating a link. You can also choose whether you want to use an existing connection file or create a new one.

Linking to a connection file When you establish a link between the connection file and a page, the page's **ConnectionFile** property is set to the name of the file. Each time you open the page, Access will read the connection file, extract the connection information, and set the **ConnectionString** property of the page.

Using a connection file without creating a link When creating a page, you can use the contents of a connection file to set the **ConnectionString** property of the page, but choose not to create a link between the page and the connection file.



[Changing the connection information of a page](#)

When a connection file is used Either change the **ConnectionFile** property of the page to point to a different connection file, or edit the connection file in a text editor. If you choose to edit the connection file, remember that all other pages that use the connection file will also be affected by the changes you make. If you edit the **ConnectionString** property of a page that is linked to a connection file, the link will be broken and the **ConnectionFile** property will be set to null.

When a connection file is not used Either create a link between the page and a connection file by setting the **ConnectionFile** property, or edit the **ConnectionString** property. When you create a link, Access will automatically set the **ConnectionString** property based on the contents of the connection file.

Create a connection file

If you want to create a connection file to link to a page that does not exist, follow the procedure for creating a data access page. Use the following procedure if you want to create a connection file to link to an existing page.

1. If a database is open in Microsoft Access, close it.
2. On the **File** menu, click **New**.
3. In the **New File** dialog box, click **Blank Data Access Page**.
4. In the **Select Data Source** dialog box, click **New Source**.
5. In the **Data Connection Wizard** dialog box, select the data source that you want to connect to.
6. Follow the directions in the wizard dialog box.

Access creates a connection file with the information you specified.

7. In the **Select Data Source** dialog box, click **Cancel**.

Access creates a blank data access page.

8. Close the data access page.

You can use the connection file with existing or new data access pages.

Specifying absolute path in the connection string of a page

If you specify an absolute path in the connection string of a data access page, such as "N:\data\Sales.mdb", the page might not be able to connect to the data through the network at a later point in time. A drive letter can vary on a computer or may not always be defined, so using absolute path is not a reliable way to specify connection information.

Make sure you use a universal naming convention (UNC) path in the connection string, such as \\Sources\Sales\Sales.mdb. A UNC path is a reliable and consistent way for the page to locate data sources on a network.

[Show All](#)

Open a data access page on a Web server in Access

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Double-click the **Edit Web page that already exists** object shortcut.
3. Under **Look in**, click **My Network Places**, and then double-click the network location that contains the [data access page](#) you want to open.
4. In the folder list, locate and double-click the folder that contains the data access page.
5. Double-click the page you want to open.

Note Make sure your supporting files folder is in the same folder as the data access page so that any links are maintained.

[Tips](#)

- To quickly open a data access page on a Web server, in the **File name** box type the address for the page you want to open, and then press ENTER. For example, you might type **http://sales/february/Page1.htm** to open a page from the sales department. If a [Web folder](#) at **http://sales/february** doesn't already exist, it will be created automatically when you type an address in the **File name** box.
- If the **Web** toolbar is displayed, you can type the address in the **Address** box, and then press ENTER to quickly open the page in your browser.

[Show All](#)

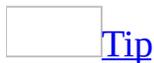
Open an Access data file on a network by using a UNC address

You might want to open a [Microsoft Access data file](#) on a network share by using a [UNC](#) path, instead of using the drive letter of a mapped network drive in Microsoft Windows Explorer. A drive letter can vary from one computer to another or may not always be defined, whereas a UNC path is a reliable and consistent way for Access to locate the Access data file.

1. On the **File** menu, click **Open** .
2. In the **File name** box, type the UNC address for the network share you want, and then press ENTER.

For example, type `\\Computer1\Files` to open an Access data file on the Files share of the Computer1 server.

3. In the folder list, locate and open the Access data file you want.



[Tip](#)

To open an Access data file on a network share you've recently opened by using a UNC address, click the arrow next to the **File name** box within the **Open** dialog box, and then click the UNC connection.

[Show All](#)

Set or change the connection information of a data access page

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. On the property sheet, click the **Data** tab.
4. Do one of the following:

[Set or edit the **ConnectionString** property of the page](#)

Note Editing the **ConnectionString** property of a page that is linked to a connection file will break the link between the page and the connection file.

- In the **ConnectionString** property box, either type the new connection string or click the **Build** button to use the **Data Link Properties** dialog box to build the connection string.

[Tip](#)

If you need more room to type the connection string, press SHIFT+F2 to open the **Zoom** box.

[Specify or change the connection file that the page is linked to](#)

- In the **ConnectionFile** property box, either type the name of the new connection file or click the **Build** button to locate the file.

Note Microsoft Access will automatically update the **ConnectionString** property to reflect the new connection information.

[Edit the connection file of the page](#)

Note If the page is linked to a connection file, you can edit the contents of the file. Your edits will affect other pages that are linked to the connection file.

1. In the **ConnectionFile** property box, click the **Build** button .
2. In the **Select Data Source** dialog box, right-click the connection file, point to **Open With**, and click **Notepad**.
3. Search for "<odc:ConnectionString>" (without the quotation marks) to locate the connection information and make the changes you want.
4. Save and close the file.
5. In the **Select Data Source** dialog box, click **OK**.

Notes

- If you have specified a default connection file for new data access pages, Microsoft Access will automatically set the **ConnectionFile** and **ConnectionString** properties for a page when you create it.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. For more information about a PivotTable list, spreadsheet, or chart, click **Help**  on that component's toolbar.

[Show All](#)

Set or change the record source of a data access page section

You can change the [record source](#) of a [section](#) on an ungrouped page, if the record source that you want to use contains all the fields that are already on the header section. For example, if you add fields from the Order Details table, and then decide to use the Order Details Extended query instead of the Order Details table, you can simply change the property of the header section.

1. Open the data access page in [Design view](#).
2. Double-click the [section bar](#) of the header section.
3. In the **RecordSource** property box, select a record source from the list.

Note The names of the sections and the **ControlSource** property of the controls will be automatically updated to reflect the new record source.

[Show All](#)

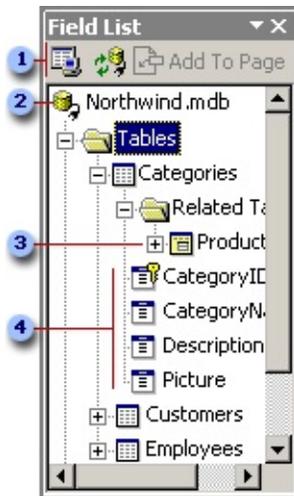
About adding data to a data access page by using the field list

When you use the [field list](#) to add data to your data access page, Microsoft Access automatically creates [controls](#) that are bound to the fields you add, just like it does with the form and report field list. However, the form and report field list and the data access page field list are not the same. Before you can add bound controls to a form or report, you must first bind the form or report to a specific record source. Once you do that, only the fields from that record source are displayed in the field list. A data access page, on the other hand, isn't bound until you add controls to it, and then it can be bound to more than one record source at a time. Therefore, the field list for a data access page displays all the fields you can choose from in the database to add to the page.

Before you can add data to a data access page, you must be connected to a [Microsoft Access database](#) or [Microsoft SQL Server database](#). When the page is connected to one of these databases, the field list displays folders containing all of the tables or [queries](#) available in the database.

[Contents of a field list](#)

In the field list, when you click the expand indicator  next to a specific table or query, you'll see the list of fields for that record source. If the record source is a table, you'll see a Related Tables folder above the list of fields. The folder lists the tables and fields that are related to the table you expanded in the field list. If you rest the pointer on a table in the Related Tables folder, you'll see the field through which the tables are related. If you add fields directly from tables that are related, or if you add fields from queries that use related tables, Microsoft Access automatically uses the existing [relationships](#) as you design the data access page. If you add fields to the page from tables that aren't related or from tables that have multiple relationships to tables already used on the page, Access will prompt you each time to define a new relationship. Relationships you define in this way are used only for the current data access page; they are not added to the database.



- 1 Buttons that display the connection information for the page, refresh the field list, and add the selected item to the current section of the page
- 2 The name of the database to which the page is connected
- 3 Table related to the Categories table
- 4 Fields in the Categories table

If you're working in a multiuser environment, you might want to refresh the field list occasionally in case others have added tables or queries since you began designing the data access page.

[Adding fields to the page](#)

You can drag a single field, multiple fields from a table or query, or an entire table or query from the field list to the page.

When you drag a single field, Microsoft Access creates the appropriate individual control based on the type of data stored in the field. You can also choose the individual control you want it to create before you drag the field to the data access page.

When you drag a table, query, or multiple fields from a table or query to the page, if the **Control Wizards** tool  in the toolbox is selected, Access displays the Layout Wizard, which asks whether you want to add fields as individual controls (in columnar or tabular layout) or as a Web control (Microsoft Office

PivotTable Component, Spreadsheet Component, or Chart Component). If you want to arrange the controls in a free-form custom layout, use columnar or tabular layout. Add fields as part of a Web control if you want to display data that you can rearrange, summarize, and otherwise analyze in different ways. If the **Control Wizards** tool is not selected when you drag multiple fields, a table or query, Access adds individual controls to the page.

Note The PivotTable list and chart have their own field lists, separate from the field list for the data access page. The field list of a component shows all of the fields you added from the data access page's field list to the component. It does not list all of the fields and record sources available in the database. If you remove a field from the PivotTable list or chart, the field remains in its field list so you can add it later. Unlike the data access page's field list, the component's field list is also available in the browser so that others using the page can add fields to the component as they're analyzing data.

[Building the page's data model](#)

As you drag fields and record sources to the data access page and add [group levels](#), change the sort order, and so on, Microsoft Access constructs a query behind the scenes that translates into [Structured Query Language \(SQL\)](#) what you've laid out on the screen and otherwise specified for the design. Using the existing [relationships](#) that are defined between record sources, Access creates a query that retrieves the data you selected and presents it the way you've specified.

If you add fields from record sources that have no relationship, Access displays the Relationship Wizard so you can define the relationship yourself.

When you add a field to the data access page, Access also adds the field to the page's data model, which is the collection of data you've added to the page. The data model, represented on the data outline, shows which record sources are used for the page and how the data is related.

[Show All](#)

Refresh the field list of a data access page

If you're working in a multiuser environment, or if you modified an underlying table or [query](#) after opening the page, you might want to refresh the [field list](#) of the page.

1. If the field list is not open, click **Field List**  on the toolbar.
2. Click **Refresh**  on the field list toolbar.

[Show All](#)

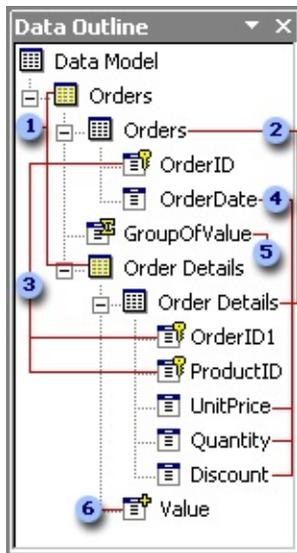
About the data outline

The data outline displays a tree view of the data model of a [data access page](#). It lists the [record sources](#), fields, and [calculated controls](#) of a page. Because the field list of a page does not show the contents specific to a page, you can use the data outline to review the structure of a page. You can also select the objects displayed in the data outline, set their properties, define and edit relationships between record sources, and delete fields and record sources.

The following illustration shows the Orders page with two group levels. The inner group level contains fields from the OrderDetails table and a calculated control named Value. In the outer group level, the header section contains fields from the Orders table and the footer section contains a calculated total control named OrderValue.

Order Date:	04-Jul-1996
Order ID:	10248
Product ID:	11
Unit Price:	\$24.00
Quantity:	12
Discount:	0.00%
Value:	\$288.00
Order Details 1 of 3	
Sum Of	\$560.00

The following illustration shows the data outline for the Orders page.



- 1 The **RecordsetDef** object, which contains one or more record sources
- 2 The underlying record sources
- 3 Primary key fields
- 4 Data fields
- 5 Aggregate field
- 6 Calculated field

[Show All](#)

Open or close the data outline

The Data Outline displays a tree view of the data model of a data access page.

1. Open a data access page in [Design view](#).
2. Click **Data Outline**  on the toolbar.

[Show All](#)

About saving an object as a data access page

You can save a table, query (select, union, or crosstab), form, or report in a [Microsoft Access database](#) as a [data access page](#). You can also save tables, views, stored procedures, functions, and forms in a [Microsoft Access project](#) as a data access page. Saving an object as a data access page enables you to quickly convert an object into a page, and allows users of your application to review, enter, and analyze data over the Internet or an intranet.

For example, if you have a sales report that you want to make available over the Web, instead of creating a data access page and customizing it to look like the sales report, you can save the report as a data access page. Using the **Save As** command on the **File** menu will save a data access page that is similar in appearance and functionality to the original report. If you want to make changes to the design of the resulting page, you can open the page in [Design view](#) and make any changes you want.

The default view—the view in which the object is open when you carry out the **Save As** command—determines the design of the data access page.

The following table lists the outcome of each view.

Default view (current view)	Outcome
Datasheet view	A page that has controls arranged as a datasheet
Form view	A page that has controls arranged as a datasheet
PivotTable view	A page that uses a Microsoft Office PivotTable Component
PivotChart view	A page that uses a Microsoft Office Chart Component

If you carry out the **Save As** command after making changes to the object's formatting, but before saving your changes, the current formatting—not the saved formatting—will be used to create the page.

[Implications of using the **Save As** command](#)

Control support You will notice the following differences after saving a form or report as a data access page:

- Bound object frames, unbound object frames, toggle buttons, and tab controls are not supported in data access pages. If the original object includes any of these controls, Microsoft Access will ignore them while creating the page.
- Diagonal lines are not supported in data access pages, so diagonal lines in a form or report will appear as horizontal lines on a data access page.
- Value lists as row sources are not supported in data access pages. Such controls will be left unbound on a page.
- Data access pages don't support displaying multiple columns in list boxes and combo boxes, so you will see only the first visible column of the original control on a page.

Subforms and subreports Subforms and subreports on a form or report are not converted when you carry out the **Save As** command. However, you can open the page in [Design view](#) and create additional [group levels](#) to make the page appear similar to the original object.

Control naming Controls on forms and reports need not have unique names, but names of controls on a page must be unique. While creating the page, Access will change control names that are not unique.

Controls in the header and footer sections Controls in the Form, Report, or page header sections will be placed as [unbound controls](#) in the caption section of the outermost group level. Controls in the Form, Report, or page footer sections will be placed as unbound controls in the navigation section of the outermost group level.

Code Code in a form or report that cannot run from a data access page is imported into the page as a comment block at the end of the document.

Pictures Pictures in a form or report are converted to bitmaps and placed in a folder named "Images".

Group level properties By default, the **AllowAdditions**, **AllowDeletions**, and

AllowEdits properties of all group levels in a data access page created from a report are set to **False**. However, you can open the page in Design view and change the property settings.

Expressions [Expressions](#) that refer to form or subform properties are ignored.

Security Note Data access pages aren't actually stored in the Access file; they are stored as HTML files either in the local file system, in a folder on a network share, or on an [HTTP](#) server. For this reason, Microsoft Access has no control over the security of data access page files. To help protect the data access page, you must employ security measures for its link and HTML file by using the file system security of the computer where you have stored these files. To help protect the data accessed by the page, you must either employ security measures for the database that the page is connected to or configure Microsoft Internet Explorer security settings to prevent unauthorized access.

About saving a password in a data access page

When you save a password in a data access page, the password is saved in its unencrypted form in the page. Users of the page will be able to view the password. A malicious user could access the information to compromise the security of the data source.

To protect your data, on the **Connection** tab of the **Data Link Properties** dialog box, clear the **Allow saving password** check box.

[Show All](#)

Save a copy of a data access page to a Web server

You can save a copy of a Microsoft Access [data access page](#) to a [Web server](#).

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Double-click the data access page that you want to open.
3. On the **File** menu, click **Save As**.
4. In the **Save Data Access Page** *<object name>* **To** box, type the name of the new data access page.
5. Under **Look in** in the **Save As Data Access Page** dialog box, click **My Network Places**.
6. Click **Add Network Place**, select either **Create a shortcut to an existing Network Place** or **Create a new Network Place**, click **Next**, and then follow the instructions in the Add Network Place Wizard.



When you save a data access page, all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. If you move or copy your page to another location, you must also move the supporting folder so that you maintain all links to your page.

[Save a copy of a data access page to a Web server in Windows Explorer](#)

[Show All](#)

Save a copy of a data access page to a Web server in Windows Explorer

1. In Microsoft Windows Explorer, right-click the file that you want to copy or move to a Web server.
2. On the shortcut menu, click **Copy**.
3. Click **My Network Places**.
4. In the list of **My Network Places**, double-click the folder that you want, and then right-click the destination folder you want to save the [data access page](#) to.
5. On the shortcut menu, click **Paste**.
6. If you don't see the Web server that you want to save your data access page to, double-click **Add Network Place** and follow the instructions in the Add Network Place Wizard.



When you save a data access page, all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. If you move or copy your page to another location, you must also move the supporting folder so that you maintain all links to your page.

[Show All](#)

Work with FTP

You can save data access pages to an [FTP](#) file server. You can also customize the way you work with FTP sites.

[Save a data access page to an FTP site](#)

To do this procedure, your company must have an [intranet](#), or you must have access to an [Internet](#) site that supports saving files. In addition, you must have access rights to save, and you must add the FTP site to your list of FTP locations.

1. If the data access page you want to save to an FTP site isn't already open, open it in [Page view](#) or [Design view](#). The page you want to save can be a new data access page that has never been saved or an existing data access page.
2. On the **File** menu, click **Save As**.
3. In the **Save As** dialog box, type a name for the link to the data access page and click **OK**.
4. In the **Save As Data Access Page** dialog box, click **FTP Locations** in the **Save In** box.
5. In the list of FTP sites, double-click the site you want, and then double-click the location you want to save the data access page to.

Note You must have write permissions for the FTP location to save the data access page there.

6. In the **File name** box, type a name for the .htm file.
7. Click **Save**.

[Add an FTP site to the list of FTP locations](#)

You can add an FTP site to the **FTP Locations** list in the **Open** and **Save As** dialog boxes. To do this, your company must have an [intranet](#) or you must have access to the [Internet](#)— for example, you could have a modem and an Internet account through a [service provider](#), or you could have access through your corporation's network.

1. On the **File** menu, click **Open**.
2. In the **Look In** box, click **Add/Modify FTP Locations**.
3. In the **Name of FTP site** box, type the FTP site name; for example, type **ftp.microsoft.com**.
4. To log on to a site anonymously, click **Anonymous** (some sites don't allow this).

To log on to a site you have user privileges for, click **User**, type your logon name, and then type your password if prompted.

5. Click **Add**.

[Remove an FTP site from the list of Internet sites](#)

1. On the **File** menu, click **Open**.
2. In the **Look In** box, click **FTP Locations**.
3. Right-click the FTP site you want to remove, and then click **Remove** on the shortcut menu.

[Change the logon name or password for an FTP site](#)

You can change the way you log on to an FTP site. To do this procedure, your company must have an [intranet](#) or you must have access to the [Internet](#)— for example, you could have a modem and an Internet account through a [service provider](#), or you could have access through your corporation's network.

1. On the File menu, click **Open**.
2. In the **Look In** box, click **FTP Locations**.
3. Right-click the FTP site you want to change, and then click **Modify** on the shortcut menu.
4. Change the options you want.

[Show All](#)

Customize a data access page in Design view

Add data

[Add fields using the field list](#)

The field list for a data access page displays all fields you can choose from in the database to add to the page. When you use the [field list](#) to add data to your data access page, Microsoft Access automatically creates [controls](#) that are bound to the fields you add, just like it does with the form and report field list.

[Calculate a value](#)

If you want to do complex calculations using spreadsheet functions, use the Microsoft Office Spreadsheet Component. For all other calculations, you can use the [bound span control](#).

[Calculate a total or other aggregate values](#)

Use the AutoSum feature to calculate aggregate values on a data access page. For numeric values, you can choose from **Sum**, **Average**, **Min**, **Max**, **Count**, **StDev**, and **Any**. For text, date, and time values, you can choose from **Min**, **Max**, **Count**, and **Any**.

Customize the appearance of a page

[Customize a record navigation toolbar](#)

You can show or hide buttons on the record navigation toolbar, change the image that is displayed on a navigation button, or make a control— such as a text box or [bound span control](#)— work like a navigation button.

[Customize an expand control](#)

You can change the picture that is displayed on an [expand control](#), or make a text box or [bound span control](#) work like an expand control.

[Apply, change, or remove a theme](#)

A theme is a set of unified design elements and color schemes for bullets, fonts, horizontal lines, background images, and other data access page elements. A theme helps you easily create professional and well-designed data access pages.

You can apply a theme to a page that does not have a theme, change the current theme, or remove a theme. Before applying a theme, you can preview a sample page that uses the theme.

Sort, group, and filter records

[Set the default sort order of records](#)

You can sort records at each [group level](#) on one or more fields. Each field can specify a separate sort order (ascending or descending).

[Group records](#)

You can group records at each [group level](#) on one more fields. You can group on individual values of a field, prefix characters, or a range of values.

[Filter records](#)

You can filter records either before or after they are retrieved from the database. Filtering records before they are retrieved from the database can improve data access page performance because there are fewer records to download from the underlying database.

[Show All](#)

Set alternate row color for a data access page

To display alternate rows in the header and footer sections of a group in a different color, the **DataPageSize** property of the group must be set to a value greater than 1.

1. Open a data access page in [Design view](#).
2. Click the arrow on the header [section bar](#) of the group for which you want to set the color, and then select **Group Level Properties**.
3. Set the **AlternateRowColor** property to the color you want.

Note The default alternate row color for headers and footers in new data access pages is whitesmoke. You can change the default setting by changing the value of the **AlternateRowColor** property on the **Pages** tab in the **Options** dialog box.

[Show All](#)

Set the section indent for a data access page

When you set the left indent for a [section](#) in a data access page, Microsoft Access automatically sets the left indent of the other sections in the same [group level](#).

1. Open a data access page in [Design view](#).
2. Double-click the [section bar](#) of a section in the group for which you want to specify the left indent.
3. Set the **MarginLeft** property to what you want.

Notes

- The default section indent for new data access pages is .25 inches (or .635 cm). To change the default setting, click **Options** under the **Tools** menu, click the **Pages** tab, and type a value in the **SectionIndent** box.
- Setting the **MarginLeft** property of a section overrides the **SectionIndent** setting. When you add or remove a group level, Access uses the default section indent setting to format the sections of all group levels on the page, except those for which you explicitly set the **MarginLeft** property.
- By default, the left indent of sections in ungrouped and unbound data access pages is set to 0.

[Show All](#)

Show or hide the record selector

[For a form](#)

1. Open the form in [Design view](#).
2. Double-click the [form selector](#) to display the property sheet.
3. Do one of the following:
 - To show the record selector, set the **RecordSelectors** property to **Yes**.
 - To hide the record selector, set the **RecordSelectors** property to **No**.

[For a group on a data access page](#)

1. Open the data access page in [Design view](#).
2. Right-click a [section](#) that belongs to the group for which you want to show or hide the record selector, and then click **Group Level Properties**.
3. Do one of the following:
 - To show the record selector, set the **RecordSelector** property to **True**.
 - To hide the record selector, set the **RecordSelector** property to **False**.

[Show All](#)

Remove the placeholder text at the top of a data access page

1. Open the data access page in [Design view](#).
2. Click **Click here and type title text**.
3. Press DELETE.

Note Even if you don't delete it, the placeholder text won't appear when you view the page in [Page view](#) or in Microsoft Internet Explorer.

[Show All](#)

About working with graphics on a data access page

Inserting graphics from a file

When you insert a graphic from a file you can either insert the graphic by copying and pasting it from another location, or you can insert a link to the graphic at a [fixed location](#), such as another [Web server](#) or to a [relative file location](#). Linking a graphic to your [data access page](#) reduces the size of the page, makes it easier to update the graphic, and makes it easier to share the graphic between different data access pages. In Microsoft Windows, if you move or copy a data access page to another location, make sure that you also move or copy any supporting files— such as graphics, background textures, and bullets— so that you maintain the graphic links to your page.

Aligning graphics

The **Alignment and Sizing toolbar** (Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later) in data access page [Design view](#) provides alignment commands that help you arrange your graphics. If you are using Internet Explorer 5.5 or later, use the **Align** and **Size** commands on the **Format** menu.

[Show All](#)

Format the body text of a data access page

1. Open the data access page in [Design view](#).
2. Select the text in the body of the page.
3. Do one or more of the following:
 - To format text as a bulleted or numbered list, click **Numbering**  or **Bullets**  on the toolbar.
 - To increase or decrease the text indent, click **Increase Indent**  or **Decrease Indent**  on the toolbar.
 - To set the font properties, click the corresponding button on the toolbar.

[Show All](#)

Add or delete an expand control

1. Open the data access page in [Design view](#).
2. Do one of the following:

[Add an expand control](#)

1. Click **Expand**  in the [toolbox](#).
2. Click in the [group header](#) where you want to place the expand control.

[Delete an expand control](#)

1. Make sure the [expand control](#) that you want to delete is selected.
2. Press DELETE.

Notes

- You can customize the appearance of the expand control by changing its property settings.
- When you delete the expand control for a group header, Microsoft Access automatically sets that group's **ExpandedByDefault** property to **True**. If you set the **ExpandedByDefault** property to **False** after deleting the expand control, you won't be able to see the records in the lower group levels.

[Show All](#)

Customize the expand control on a page

1. Open the data access page in [Design view](#).
2. Scroll to the header section that has the expand control  that you want to customize.
3. Do one or more of the following:

[Change the picture that is displayed on the expand control](#)

- Do one of the following:

[Select from a list of images included in Microsoft Access](#)

1. Double-click the expand control .
2. On the **Other** tab, in the **Src** property box, select the item you want.

[Use custom images](#)

To change the picture on a navigation button, you need two picture files to represent the expanded status and collapsed status of the button.

1. Set the name of the file that contains the expanded image to the name of the file that contains the collapsed image followed by the string "expanded".

For example, if the collapsed picture file is named folder.gif, name the expanded picture file folderexpanded.gif.

2. Double-click the button whose picture you want to change.
3. On the **Other** tab in the property sheet, in the **Src** property box, type the path and name of the collapsed picture file.

At run time, Microsoft Access will automatically show the contents of the expanded picture file when the group is expanded.

[Make a control work like an expand control](#)

1. If the control that you want to use as the expand control is not already on the page, add the control to the page using the [toolbox](#).
2. Double-click the control to display its property sheet.
3. On the **Other** tab, in the **ClassName** property box, append the existing property value with a space and the string **MsoExpandCollapse**.

For example, if the property is set to **MsoBoundSpan**, set it to **MsoBoundSpan MsoExpandCollapse**.

Notes

- The control will act as a toggle button but will not indicate the expanded or collapsed state.
 - When using custom expand controls, you might want to provide custom help or instructions on your page to assist users.
4. Make sure the **ExpandedByDefault** property of the group level that has the customized expand control is set to **False**.

[Show All](#)

Add or delete a record navigation control

1. Open the data access page in [Design view](#).
2. Do one of the following:

[Add a record navigation control](#)

When you add a [record navigation section](#) to a [group level](#), Microsoft Access automatically adds a [record navigation control](#). Use this procedure for a group level that has a record navigation section but doesn't have the record navigation control.

1. Click **Record Navigation**  in the [toolbox](#).
2. Click the record navigation section where you want to place the record navigation control.

[Delete a record navigation control](#)

1. Click the [record navigation control](#) that you want to delete.
2. Press DELETE.
3. Click **Yes** to set the **DataPageSize** property of the group level to **All**.

Notes

- If you don't want to show the record navigation control for a group level, set the **RecordNavigationSection** property of the group level to **False**.
- You can customize a record navigation control by changing its property settings.

[Show All](#)

Customize a record navigation toolbar

1. Open the data access page in [Design view](#).
2. Scroll to the [record navigation section](#) whose toolbar you want to customize.
3. Do one or more of the following:

[Show or hide a navigation button](#)

1. Right-click the navigation toolbar, and then click **Navigation Buttons** on the shortcut menu.
2. Do one or both of the following:
 - Select the items that you want to include in the toolbar.
 - Clear the items that you want to hide from the toolbar.
Alternatively, you can click the button on the toolbar, and then press DELETE.

Note If you don't see a check box to the left of the item, it means the item is not available to the page.

[Change the picture that is displayed on a navigation button](#)

To change the picture on a navigation button, you need two picture files to represent the active status and inactive status of the button.

1. Set the name of the file that contains the inactive image to the name of the file that contains the active image followed by the string "inactive".

For example, if the active picture file is named Next.gif, name the inactive picture file Nextinactive.gif.

2. Double-click the button whose picture you want to change.
3. On the **Other** tab in the property sheet, in the **Src** property box, type the path and name of the active picture file.

At run time, Microsoft Access will automatically show the contents of the inactive picture file when the button becomes inactive.

 [Make a control \(such as a text box or bound span control\) work like a navigation button](#)

1. If the control that you want to use as a navigation button is not already on the page, add a control to the page using the [toolbox](#). You can add the control to the [header](#), [footer](#), or [caption section](#) of a [group level](#).
2. Double-click the control to display its property sheet.
3. On the **Other** tab, in the **ClassName** property box, append the existing property value with a space and one of the following class names:

For this button Set the ClassName property to

First	MsoNavFirst
Previous	MsoNavPrevious
Recordset Label	MsoNavRecordsetLabel
Next	MsoNavNext
Last	MsoNavLast
New	MsoNavAddNew
Delete	MsoNavDelete
Save	MsoNavSave
Undo	MsoNavUndo
Sort Ascending	MsoNavSortAsc
Sort Descending	MsoNavSortDesc
Filter By Selection	MsoNavFilter
Filter Toggle	MsoNavToggleFilter
Help	MsoNavHelp

Notes

- When you add custom navigation buttons or change the picture on the navigation buttons, you might want to provide custom help or instructions on your page to assist users.
- Each navigation button has its own property sheet. Double-click the button to display it.
- If you want to change the behavior of the toolbar on the **onmouseover** and

onmousedown events, open the Microsoft Script Editor and edit the styles of the `MsoNavButtonMouseDown` and `MsoNavButtonMouseOver` classes. You can also edit the styles for `MsoNavContainer` and `MsoNavButton` classes to control the appearance of the toolbar and navigation buttons.

[Show All](#)

Change the aggregate function of a calculated control

1. Open the data access page in [Design view](#).
2. Double-click the [calculated control](#) whose function you want to change.
3. On the **Data** tab, in the **TotalType** property box, select a function from the list.

Notes

- When you change the [aggregate function](#) of a control, the default alias and caption of the control change to reflect the underlying aggregate function. To prevent the control name from changing, type a new alias for the control in its **ControlSource** property. User-specified aliases don't change when the aggregate function is changed.
- Changing the aggregate function of a control does not affect other aggregate controls, such as those that calculate subtotals or grand totals.

[Show All](#)

Specify whether a group is expanded or collapsed by default on a page

When you set the **ExpandedByDefault** property, you're specifying whether to show or hide the records in the next lower [group level](#) when the page is displayed in [Page view](#) or in Microsoft Internet Explorer.

1. Open the data access page in [Design view](#).
2. Right-click the [section bar](#) of a section that belongs to the group whose settings you want to change, and then click **Group Level Properties**.
3. In the **ExpandedByDefault** property box, do one of the following:
 - To open the page with the records in the next lower group level displayed, select **True**.
 - To open the page with the records in the next lower group level hidden, select **False**.

Note The page will load faster in Page view and in Internet Explorer if the **ExpandedByDefault** property is set to **False** for the highest group level on the page.

[Show All](#)

Specify default settings for new data access pages

1. Open a data access page in [Design view](#).
2. Do one or more of the following:

[Set the default connection information](#)

1. On the **Tools** menu, click **Options**, and then click the **Pages** tab.
2. Select the **Use Default Connection File** check box.
3. In the **Default Connection File** box, type the name of the connection file that you want to use for all new pages. Or click the **Browse** button to locate the file.

Note Setting the **ConnectionFile** property of the page overrides the default connection file setting.

[Set the default indent for sections on grouped pages](#)

1. On the **Tools** menu, click **Options**, and then click the **Pages** tab.
2. In the **Section Indent** box, type the value you want.

Note Setting the **MarginLeft** property of a section overrides the default section indent setting. When you add or remove a [group level](#) in a page, Microsoft Access uses the default section indent setting to format the sections of all group levels on the page, except those groups for which you explicitly set the **MarginLeft** property.

[Set the default alternate row color for groups](#)

1. On the **Tools** menu, click **Options**, and then click the **Pages** tab.
2. In the **Alternate Row Color** box, type the value you want. Or click the

Build button to select the color you want.

Notes

- Setting the **AlternateRowColor** property of a [group level](#) overrides the default alternate row color setting.
- When you apply a [theme](#) to a page, the alternate row color setting of the group levels in the page is ignored.

[Set the default styles for caption and footer sections](#)

1. On the **Tools** menu, click **Options**, and then click the **Pages** tab.
2. Do one or both of the following:
 - To specify the style for caption sections, type the settings you want in the **Caption Section Style** box.
 - To specify the style for footer sections, type the settings you want in the **Footer Section Style** box.

For more information about how to specify styles, see Microsoft Internet Explorer Help.

[Set the default theme](#)

1. On the **Format** menu, click **Theme**, and then select the theme you want to use in new pages.
2. Select or clear the check boxes below the list of themes to customize the theme.
3. Click **Set Default**.
4. In the message box that appears, click **Yes** to confirm.

[Show All](#)

Set the default control type for a data access page

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. In the property sheet, click the **Data** tab.
4. In the **DefaultControlType** property box, select **Text Box** or **Bound Span**.

Notes

- If the first control that you add to the page is a [bound span control](#), the **DefaultControlType** property is automatically set to **Bound Span**.
- Changing the default control type does not affect the controls that are already on the page.
- To override the default control type setting for a field, make sure the tool that you want is selected in the [toolbox](#), and then add the field to the page.
- Using bound span controls improves the performance of the page.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Show All](#)

Set the bound span as the default control for a data access page

Using [bound span controls](#) on a data access page makes the page load faster in [Page view](#) or in Microsoft Internet Explorer. However, you can't edit the values displayed in bound span controls.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

1. Open the data access page in [Design view](#).
2. On the **Edit** menu, click **Select Page**.
3. Set the **DefaultControlType** property to **Bound Span**.

[Show All](#)

Specify which record source of a page supports edits

When a data access page includes fields from multiple tables, or from a [query](#) that includes fields from multiple tables, you can choose the table whose fields can be edited in [Page view](#) or in Microsoft Internet Explorer.

1. Open the data access page in [Design view](#).
2. Double-click the header section that contains the editable fields.
3. On the **Data** tab, in the **UniqueTable** property box, select the table you want.

[Show All](#)

Define or edit a relationship between record sources in a data access page

If you add fields to the data access page and there isn't an existing [relationship](#) between the tables or [queries](#) they came from, Microsoft Access displays the Relationship Wizard. Relationships you define using the wizard apply only to the current data access page. They are not added to the database; therefore, they won't appear in the [Relationships window](#) in a [Microsoft Access database](#) or the [database diagram](#) window in a [Microsoft Access project](#).



[Define a relationship](#)

1. In the **Relationship Wizard** dialog box, indicate which field you want to use to relate the [record source](#) to the data access page and the record sources that are currently part of the data access page's [data definition](#).
2. Select the option that indicates the type of relationship you want between the two record sources.



[Edit an existing relationship](#)

1. If the [data outline](#) window is not displayed, click **Data Outline** on the **View** menu.
2. In the Data Outline window, locate the tables or queries whose relationship you want to edit. Look for entries with the page row source  icon to their left.
3. Right-click the nested entry, and then click **Relationships**.
4. In the **Relationship Wizard** dialog box, indicate the fields that you want to use to relate the two objects.
5. Select the option that indicates the type of relationship you want between the two [record sources](#).

Note If a relationship type is unavailable at the bottom of the wizard dialog box, it means Microsoft Access determined that only one type of relationship is possible between the two record sources.

[Show All](#)

Set the encoding for a Web page

When you are authoring a data access page in [Design view](#), you can specify the [encoding](#) that a Web browser will use to display the page.

1. Open the data access page in Design view.
2. On the **Tools** menu, click **Options**.
3. Click the **General** tab, and then click **Web Options**.
4. In the **Web Options** dialog box, click the **Encoding** tab.
5. In the **Save this document as** list, click the language you want.

Note If you want Access to always save your pages using a default language encoding, select the **Always save Web pages in the default encoding** check box. This setting affects the current page and future pages that you save. It is useful if you reuse pages from other sources and want to store every page in one encoding.

[Show All](#)

Work with data in Page view

[Add, edit, and delete data](#)

[Add a record](#)

1. On the record navigation toolbar, click **New** .

Note If you don't see the **New** button, it means one of the following:

- The page does not support data entry.
 - The designer of the page has customized the button. Look for custom instructions on how to use the page, or contact the designer.
2. Type the data you want in the first field, and then press TAB to go to the next field.
 3. When you are finished adding data to each field, save the record.

Note If you make a typing mistake, press BACKSPACE. To cancel your changes in the current field, press ESC. To cancel your changes to the entire record, click **Undo**  on the record navigation toolbar.

[Edit the data in a field](#)

1. Click in the field you want to edit.

Note You cannot edit group fields and [calculated fields](#). If you can't insert the input cursor inside any of the fields in the page, the page does not support edits.

2. Type the text you want to insert.
3. Save the record.

Note If you make a typing mistake, press BACKSPACE. To cancel your changes in the current field, press ESC. To cancel your changes to the entire record, click **Undo**  on the record navigation toolbar.

[Delete a record](#)

Important Once you delete a record, you cannot undo the deletion from the database.

- On the record navigation toolbar, click **Delete** .

Note If you don't see the **Delete** button, it means one of the following:

- The page does not support deletions.
- The designer of the page has customized the button. Look for custom instructions on how to use the page, or contact the designer.

[Add, change, or delete the path to a picture](#)

Note The picture is not stored in the database. Rather, the [path](#) to the picture's location is stored in the database, and the picture is stored in a computer's file system.

1. Click inside the path field.
2. Do one of the following:
 - To add the path to the picture, type the path of the picture's location.
 - To change the path to the picture, edit the path.
 - To delete the path to the picture, select the contents of the field and press DELETE.

Notes

- You cannot edit a field on a page unless the page supports edits.
- The path can be a relative path, a path on your computer, a [universal naming convention \(UNC\)](#) path, or a [uniform resource locator \(URL\)](#). The type of path you should use depends on the requirements of the database.
- If you want to delete the picture file, you must do so from its location in your computer's file system. See Microsoft Windows Help for more information about deleting files.

[Save or undo changes](#)

[Save a record](#)

- On the record navigation toolbar, click **Save** .

Note If you don't see the **Save** button or if it appears disabled, it means one of the following:

- The designer of the page has customized the button. Look for custom instructions on the page on how to save your changes.
- The page is read-only and does not support additions or edits.

[Undo changes when adding or editing records](#)

- Click **Undo**  on the record navigation toolbar.

Note If you don't see the **Undo** button or if it appears disabled, it means one of the following:

- The designer of the page has customized the button. Look for custom instructions on the page on how to undo your changes.
- The page is read-only and does not support additions or edits.

[Navigate between fields and records](#)

[Select fields and records](#)

The following table lists mouse techniques for selecting data or records in [Page view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data
An entire field	Three times within the field
A record	A field that belongs to the record

The following table lists keyboard techniques for selecting data or records in Page view.

**To
select**

Do this

Data in a field Move the insertion point to the start of the text you want to select, and then hold down SHIFT and press the arrow keys to move to the end of the selection.

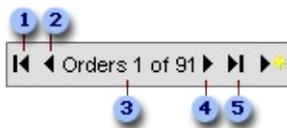
An entire field Move the insertion point to anywhere in the field, press HOME, and then press SHIFT+END.

A record Press TAB until a field that belongs to the record gets focus.

Note Your data access page might have a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component, which have their own methods of selection. To display Help for one of these components, click **Help**  on its toolbar.

 [Move between records by using navigation buttons](#)

- Use the navigation buttons located on the record navigation toolbar to quickly move between records or a group of records on a [grouped data access page](#).



- 1 First record
- 2 Previous record
- 3 Record source and number
- 4 Next record
- 5 Last record

Note If you don't see the record navigation toolbar on your page, it means one of the following:

- All the records are displayed on the page, so the navigation buttons are not necessary.
- The designer of the page has customized the toolbar. Look for custom instructions on how to use the page, or contact the designer.

 [Sort, find, and filter data](#)

 [Sort data](#)

You can sort data on a data access page, including data that's grouped. For grouped data access pages, you can sort the list of groups or the records within a group.

1. Open a data access page in [Page view](#).
2. Do one of the following:
 - In an ungrouped page, click the field you want to use for sorting records.
 - In a grouped page, expand the group that you want to sort, and then click the field you want to sort records on.

Note When you sort records within a group, the sort order applies only to that group. It does not affect the order of records in other groups.

3. Do one of the following:
 - To sort in ascending order, click **Sort Ascending** .
 - To sort in descending order, click **Sort Descending** .

 [Tip](#)

If the records are displayed in tabular format, simply click the label of the field you want to sort records on. The records will display in ascending order. To sort records in descending order, click the label again.

Notes

- When you close the data access page, you'll lose the sort order. You can sort the records again when you open the page.
- The order in which numbers, text, and special characters are sorted depends

on the database and your regional settings in Microsoft Windows Control Panel.

- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be sorted independently from the rest of the data on the data access page, and may also support additional sorting capabilities. For more information, click **Help**  on the toolbar for the PivotTable list, spreadsheet, or chart, and then search for "sorting."

[Find specific data by using a filter](#)

1. Open the data access page in [Page view](#).
2. If the records are grouped on the page, expand the group that you want to filter.
3. Move focus to the field that has the value that you want to use to filter records.
4. Click **Filter By Selection**  on the record navigation toolbar for that group.

Note On a grouped page, the filter applies only to the current group. In other words, if a page groups customer orders by country/region, and you filter the records for Brazil that have a required date of 6/2/98, that filter applies only to the records for Brazil, not to the records for the other countries/regions.

[Apply an existing filter to data](#)

You can reapply a filter that you created during the current working session.

1. Open a data access page in [Page view](#).
2. Do one of the following:
 - On an ungrouped page, click **Filter Toggle**  on the record navigation toolbar so that it is selected.
 - On a [grouped page](#), expand the group that you want to filter, and then click **Filter Toggle** on the record navigation toolbar for that group so that the button is selected.

Note This procedure doesn't apply to a Microsoft Office PivotTable Component,

Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Remove a filter to show all records](#)

1. Open a data access page in [Page view](#).
2. Click **Filter Toggle**  on the record navigation toolbar so that it is not selected.

Notes

- Removing a [filter](#) doesn't delete it. You can reapply a filter during the current working session by clicking the **Filter Toggle** button on the record navigation toolbar so that it is selected.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Use keyboard shortcuts](#)

Note To use the following keys, press the TAB key repeatedly until the [expand indicator](#) or record navigation toolbar has the [focus](#). If a [PivotTable list](#) or [spreadsheet](#) control currently has the focus, press CTRL+TAB.

SPACE or ENTER

Expand or collapse a group of records corresponding to the expand indicator in a [grouped data access page](#)

TAB or SHIFT+TAB

Select a button on the record navigation toolbar

SPACE or ENTER

Activate the selected button on the record navigation toolbar

[Show All](#)

Get help on working with a data access page in Page view

This topic explains how to view and work with a [data access page](#) in general; however, the authors of the page that you are viewing might have provided Help information specific to that page. For example, they might have added instructions directly to the page, or they might have added a custom Help button or hyperlink to information about the page.

A data access page may also include features that have their own built-in help. For example, the Microsoft Office PivotTable Component, Spreadsheet Component, and Chart Component have their own help, which explains how to work with just that feature of the page.

In [Page view](#) of a data access page, do one of the following:

- To get help for a PivotTable list, spreadsheet, or chart, click **Help**  on that component's toolbar.
- To redisplay Microsoft Access Help, close the Help window for the PivotTable list, spreadsheet, or chart, and then click **Help**  on the toolbar at the top of the Page view window.

Note Other features, such as Windows Media Player, also have their own help, which is available by right-clicking that feature.

[Show All](#)

Add, change, or delete the path to a bound picture in Page view

Note The picture is not stored in the database. Rather, the [path](#) to the picture's location is stored in the database, and the picture is stored in a computer's file system.

1. Click inside the path field.
2. Do one of the following:
 - To add the path to the picture, type the path of the picture's location.
 - To change the path to the picture, edit the path.
 - To delete the path to the picture, select the contents of the field and press DELETE.

Notes

- You cannot edit a field on a page unless the page supports edits.
- The path can be a relative path, a path on your computer, a [universal naming convention \(UNC\)](#) path, or a [uniform resource locator \(URL\)](#). The type of path you should use depends on the requirements of the database.
- If you want to delete the picture file, you must do so from its location in your computer's file system. See Microsoft Windows Help for more information about deleting files.

Show or hide records in Page view

1. Scroll to the group for which you want to show or hide detail records.
2. If the detail records are not visible, click **Expand**  next to the group heading.

If the detail records are visible, click **Collapse**  next to the group heading.

Note The **Expand** and **Collapse** buttons might look different on your page.

[Show All](#)

About Microsoft Office Web Components

Microsoft Office Web Components is a set of ActiveX controls installed with either Microsoft Office 2003 or any Office 2003 application. With Office Web Components installed on your computer, you can interactively use a [spreadsheet](#), a [PivotTable list](#), and a [chart](#) on a Web page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

If Office Web Components is not installed on your computer but you have a valid Office license that permits intranet distribution, you can configure the components so that users will be prompted to download Office Web Components from your corporate intranet. This allows users at your site to use a data access page without having Office applications installed on their computers.

For information about how to configure Office Web Components, see the *Microsoft Office 2003 Resource Kit*.

[Show All](#)

About installing, licensing, and distributing Office Web Components

When you design a Web page that uses Microsoft Office Web Components using Microsoft Office Access 2003, any user who has Microsoft Office XP Web Components installed on their computer can interact with the components in the browser to the level of interactivity you provide. That is, a user can make changes to data in a spreadsheet, change formatting, drag fields in a chart or PivotTable List, and so on, as long as you didn't protect these options at design time. Users with a valid Office license can also create, design, and modify components in a design environment such as Microsoft FrontPage.

Users who do not have a valid Office license can view the components and the data in them, and can print the view of the components, but they cannot interact with the components or manipulate them in a design environment. This means that if you distribute a data access page that uses components, users who have Office XP Web Components will have access to all functionality provided, but others can only view the data and information you've provided.

 [Installing the Office Web Components](#)

Microsoft Office Web Components are installed with Microsoft Office 2003 and Microsoft Office applications, or they can be installed separately from an installation point provided by the Web page designer. Access 2003 installs Office XP Web Components, whereas Office 2003 and other Office applications install Office 2003 Web Components.

When you distribute a Web page that uses components, users who do not have the components installed will be prompted to install the components, provided the Web page designer configures the components to do so. The Web page designer must also provide an installation point for the components and a pointer to that location on the component installation page. For more information, see the *Microsoft Office 2003 Resource Kit*.

 [Details on licensing and functionality](#)

Office 2003 installed on computer Users have full functionality and interactivity with components, including run-time and design-time capabilities.

Office 2003 application installed on computer Users can interact with the component in design mode in that application only (not in the browser or in other applications).

Office 2003 site license (user doesn't have Office 2003 installed on computer, but user's organization has an enterprise or site license agreement) Users have full functionality and interactivity with components, including run-time and design-time capabilities. However, the Web page designer must provide a location from which components can be downloaded, and must reference the site license in a license package file (.lpk) that is associated with one or more Web pages. You use the License Package Authoring Tool to create an appropriate license file for pages. For more information, see MSDN Online.

No Office 2003, Office 2003 applications, or site licenses Users can view and print the components in view-only mode, but they cannot interact with the components or use the design capabilities.

[Show All](#)

About View-only mode for Office Web Components

If you do not have a valid Office license or have not installed Microsoft Office 2003, and you do not have the correct version of Microsoft Office Web Components installed on your computer, you can view a Microsoft Office Web Component— a [Spreadsheet Component](#), [Chart Component](#), or [PivotTable Component](#)— on a Web page, but you cannot interact with it.

To be able to interact with and use full functionality of the component you are viewing, you must install Microsoft Office 2003 or have access to a valid Office license, and install the correct version of Microsoft Office Web Components on your computer. If you are in a corporate or group environment, check with your network administrator for assistance.

When you are in View-only mode, you can:

- View data.
- Print.
- Select and scroll.
- Use hyperlinks.
- Resize a component.
- Select sheets in the Spreadsheet Component.
- Expand or collapse members in a PivotTable list.
- Refresh data in a PivotTable list.
- Use the **About**, **Help**, and **Refresh** commands. Help topics that deal with interactivity (such as formatting cells, dragging fields, sorting, filtering, and so on) do not apply.

All commands except **About**, **Help**, and **Refresh** are unavailable in View-only mode.

[Show All](#)

Create a PivotTable list on a data access page

1. In [Design view](#) of a data access page, click **Toolbox**  on the toolbar.
2. In the [toolbox](#), click the **Office PivotTable** tool .
3. Click the data access page where you want the upper-left corner of the control to be.
4. To choose a data source for the PivotTable list, do one of the following:

[Use the same data source as the page](#)

- Display the data access page's [field list](#), and then drag to the PivotTable list the table or [query](#) you want to use for the PivotTable list. You can also drag individual fields from these [record sources](#) one at a time.

When you create a PivotTable list by dragging fields from the field list, Microsoft Access automatically sets the **Data source control** option on the **Data Source** tab of the PivotTable list's **Commands and Options** dialog box to **MSODSC**, which means that Access uses the same data source control for the PivotTable list as for the page itself. It also sets the **Data member, table, view, or cube name** option to the name of the table or query on which you based the PivotTable list.

[Use other data source controls that are available on the page](#)

1. Click inside the PivotTable list to activate it.
2. Right-click the PivotTable list and then click **Commands and Options**.
3. On the **Data Source** tab, select **Data source control**, and then select the control that you want from the list.
4. Under **Use data from**, click **Data member, table, view, or cube name**, and then select the element you want from the list.

[Use an external data source](#)

1. Click inside the PivotTable list to activate it.
2. Right-click the PivotTable list, and then click **Commands and Options**.
3. On the **Data Source** tab, select **Connection**, and then click **Edit**.
4. Locate the data source you want to use in the **Look in** box, and double-click it in the list. If the data source you want to use is not available, click **New Source** and follow the instructions in the **Data Connection Wizard** to create a new connection file for your data.
5. Under **Use data from**, do one of the following:
 - To use the data from a specific [record source](#) within the database, click **Data member, table, view, or cube name**, and then select the element you want from the list.
 - To use a [query](#) or command— such as an SQL statement— to select the data, click **Command text or SQL**, and then type the query or command in the box. For information about the query syntax, see the documentation for the external data source.

Security Note Under some circumstances, a user can view the information used to connect to an external data source. This may represent a security risk. For more information, see [About making connections to external data sources more secure](#).

5. Finish the PivotTable list by customizing the layout and using the toolbar and the **Commands and Options** dialog box in the PivotTable list to add totals, set properties, and so on, until the control looks and behaves the way you want.

For more help on how to design and use a PivotTable list, click **Help**  on the PivotTable list toolbar. If the toolbar is not visible, select the **Toolbar** check box on the **Behavior** tab of the **Commands and Options** dialog box.

Notes

- To create a PivotTable list that uses data from a Microsoft Excel worksheet, you have to publish the data to a Web page from Excel. If you want to turn the Web page into a data access page, open it in Microsoft Access, and then add controls that are bound to Microsoft Access or Microsoft SQL Server database data.
- When you have a Microsoft Office Web Component bound to an external

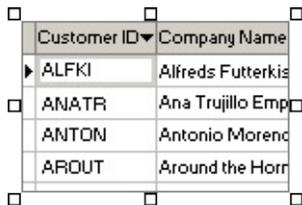
data source on the same data access page with controls bound to an Access or SQL Server database, the component control uses a different [data source control](#) than the rest of the page.

- You can create a chart that's linked to the PivotTable list so that when you change the PivotTable list, the chart updates automatically to reflect the changes.
- When you add a PivotTable list to a [grouped data access page](#), a PivotTable list appears for each unique group and contains only those records for that group. For example, on a page grouped by year with a PivotTable list containing employee sales data, there is a PivotTable list for each year, containing only the sales data for that year. You can't display the PivotTable list for more than one group at a time.

Activate a PivotTable list, spreadsheet, or chart on a data access page

Activating one of the Microsoft Office Web Components is not the same as selecting the control. When you select the control, you're selecting the outer control. Once the outer control is selected and has the focus, you can move the focus inside the control to activate its contents. When its content is activated, you can set properties and work with the features of the control. For example, in a PivotTable list, you can move columns and rows, or work with its layout and data.

1. Click the control once to select the outer control.



Customer ID	Company Name
ALFKI	Alfreds Futterkies
ANATR	Ana Trujillo Emp
ANTON	Antonio Morenc
AROUT	Around the Horr

2. Click the control again to activate the content in the control. When the content is activated, the border of the control has diagonal slashes around it.



Customer ID	Company Name
ALFKI	Alfreds Futte
ANATR	Ana Trujillo E
ANTON	Antonio Moi

Once the content is activated, you can select different parts of the control to set properties for the selected part.

[Show All](#)

Change the external connection information for a PivotTable list

1. Click the PivotTable list to select it.
2. Click inside the PivotTable list to activate it.
3. Right-click the PivotTable list, click **Commands and Options**, and then click the **Data Source** tab.
4. Click **Connection**, and then click **Edit**.
5. Locate the data source you want to use in the **Look in** box, and double-click it in the list. If the data source you want to use is not available, click **New Source** and follow the instructions in the **Data Connection Wizard** to create a new connection file for your data.

Security Note Under some circumstances, a user can view the information used to connect to an external data source. This may represent a security risk. For more information, see [About making connections to external data sources more secure](#).

6. Under **Use Data from**, do one of the following:
 - To select a different table or [query](#), click **Data member, table, view, or cube name**, and then select the [record source](#) you want from the list.
 - To change the command text, click **Command text or SQL**, and then enter the query to select the data. For information about the query syntax, see the documentation for the external data source.

Notes

- To use data from a Microsoft Excel worksheet, you have to publish the data to a Web page from Excel. If you want to turn the Web page into a data access page, open it in Microsoft Access, and then add controls that are bound to Access or Microsoft SQL Server database data.
- When you have a Microsoft Office Web Component bound to an external data source on the same data access page with controls bound to an Access or SQL Server database, the component control uses a different [data source control](#) than the rest of the page.

[Show All](#)

Create a chart for the Web on a data access page

1. Open a data access page in [Design view](#).
2. Click the **Office Chart** tool  in the toolbox, and then click and drag on the data access page until the chart is the size you want.
3. If the control isn't already selected, click it once to select it. Then, with the control selected, click it again to display the **Commands and Options** dialog box.
4. On the **Data Source** tab, specify connection information.

For more information about specifying settings, click **Help** .

5. Close the **Commands and Options** dialog box.
6. If the field list isn't visible, click **Field List**  on the **Page Design** toolbar.
7. Drag the fields you want to the appropriate location: the **Series Area**, **Category Area**, **Data Area**, or **Filter Area**.
8. Customize the chart.

For more information, select the chart, right-click it, and then click **Help**.

[Show All](#)

Add a spreadsheet to a data access page

You can add a Microsoft Office Spreadsheet Component to a data access page to provide some of the same capabilities you have in a Microsoft Excel worksheet. You can enter values, add formulas, apply filters, and so on. Use the procedure below to create a spreadsheet in which you enter raw data, or into which you import data from a Web page or text file. If within the spreadsheet you want to use data from other controls on the data access page, you need to refer to those controls in the appropriate spreadsheet cells.

1. In [Design view](#) of a data access page, click **Toolbox**  on the toolbar.
2. In the [toolbox](#), click the **Office Spreadsheet** tool .
3. Click the data access page where you want the upper-left corner of the control to be.
4. Click inside the control to activate the spreadsheet.
5. Right-click inside the control, and then click **Commands and Options**.
6. Using the **Commands and Options** dialog box, customize the appearance and behavior of the spreadsheet.

For more help on designing and using a spreadsheet, click **Help**  on the spreadsheet toolbar. If the toolbar is not visible, select the **Toolbar** check box on the **Workbook** tab of the **Commands and Options** dialog box.

[Show All](#)

Make a spreadsheet look like a text box

By hiding the component's spreadsheet features, adjusting the color of cell borders to match other controls on the data access page, and sizing the control to a cell or range of cells, you can make the spreadsheet look like the text box controls on the page.

1. Open the data access page in [Design view](#).
2. Activate the spreadsheet.
3. Right-click the control, and then click **Commands and Options**.
4. On the **Sheet** tab, clear the following check boxes: **Row headers**, **Column headers**, and **Gridlines**.
5. On the **Workbook** tab, clear the following check boxes: **Horizontal scrollbar**, **Vertical scrollbar**, **Sheet selector**, and **Toolbar**.
6. On the **Advanced** tab, clear the **AutoFit Spreadsheet** check box.
7. Resize the control by dragging the selection handles.
8. On the **Sheet** tab, set **Viewable range** to the cells you want to display. For example, to display the value in cell A1, set the **Viewable range** to **A1**.
9. Select the cells whose borders you want to change.
10. On the **Format** tab, in the **Cell format** group, set the border options to match the other controls on the data access page.

[Show All](#)

About offline data (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You don't always have online access to your company databases. However, you often want to take selective data offline (that is, download the data to a computer, such as a laptop or home computer), review and update the data away from your workplace, and then conveniently update and reconcile data changes when you have online access again. Using a combination of Microsoft products, a developer and database administrator can create an application that enables offline data access for users.

[Why take data offline?](#)

There are many reasons for working with offline data. Here are two common ones:

- You are a salesperson who is traveling and meeting customers. Before you travel, you want to review the latest sales orders and marketing trends of the customers you are visiting. This information is stored in your company's orders and customers databases. You download data about these customers to your laptop computer. While visiting these customers, you update the downloaded customer information such as phone numbers, e-mail addresses, and personnel contacts on your laptop. You also enter new orders on the laptop, but you want to fulfill these orders as quickly as possible. At the end of the day, you connect to your company's internet through a secure dial-up connection and update the orders and customers databases. Data conflicts are automatically reconciled in the orders and customers databases based on established business procedures. Then you print and fax a confirmation report for the customer to read, or e-mail a [report snapshot](#) that the customer displays using [Snapshot Viewer](#).
- You are a software developer and you want to finish up working on your company's bug tracking application at home where you don't have convenient or secure access to your corporation intranet. You take offline only the bugs assigned to you on your laptop computer. You undock your

laptop computer, go home, and then change status fields, update assignments, and so on at home. The next day, you dock your laptop computer and easily synchronize the changes with your company's bug tracking application.

[System requirements for using offline data](#)

You need the following software to create an use offline data:

- Microsoft Access 2002 or later installed on a client computer, such as a laptop or home computer.
- Microsoft SQL Server 2000 Desktop Edition installed and running on a client computer.
- Microsoft SQL Server 2000 installed and running on a remote server.
- Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later installed on the client computer and remote server.

[Creating and using an offline data application](#)

The general process of creating and using an offline data application works like this:

A developer creates a Web-based database application using one or more data access pages

In a [Microsoft Access Project](#) connected to a Microsoft SQL Server 2000 database on the remote server, a developer creates one or more data access pages that access the data the user wants to take offline. A developer needs to set the following Microsoft Office Data Source Component (MSODSC) properties to configure the Page for offline use:

OfflinePublication

OfflineCDF

OfflineType

OfflineSource

For more information on using these MSODSC properties, press F1 in the Page

property sheet on the property field.

If the offline application uses two or more Pages, or a Page contains additional images and controls, then a developer needs to create a Channel Definition Format (CDF) file and link the CDF file to the Page using the **OfflineCDF** property. Internet Explorer reads this CDF file to download all pages and HTML-related files.

The following examples shows a sample CDF file and a LINK element HTML tag added to the Page:

```
--applicationfilename.cdf
```

```
<?XML VERSION="1.0" ENCODING="windows-1252"?>
```

```
<CHANNEL HREF="default.htm"
  PRECACHE="YES"
  LEVEL="0">
  <ITEM HREF="Products.htm"></ITEM>
  <ITEM HREF="Categories.htm"></ITEM>
  <ITEM HREF="CategoryDetail.htm"></ITEM>
  <ITEM HREF="Orders.htm"></ITEM>
  <ITEM HREF="OrderDetails.htm"></ITEM>
  <ITEM HREF="xmldata\Categories.XML"></ITEM>
  <ITEM HREF="bitmaps\CorpLogo.gif"></ITEM>
  <ITEM HREF="bitmaps\ShippedItem.gif"></ITEM>
  <ITEM HREF="bitmaps\BackorderedItem.gif"></ITEM>
  <ITEM HREF="bitmaps\Discontinued.gif"></ITEM>
</CHANNEL>
```

```
--HTML tag in data access page
```

```
<link rel = "offline" href = "applicationfilename.cdf">
```

For more information on CDF, see the Microsoft Developer Network (MSDN) Web site.

A database administrator defines replication on the remote SQL Server 2000 database

A database administrator uses SQL Server 2000 to define replication on the remote SQL Server 2000 database, and create a single, named merge publication. SQL Server has a Publication Wizard which makes it easy to create a publication. The SQL Server Desktop Engine installed and running on the local computer is defined as the location of the local subscription database.

To handle potential data conflicts when the offline data is brought back online by a user, a database administrator must also define merge replication rules in the named, merge publication. This named publication can reference any database objects.

Caution It is important to define server filter restrictions in the publication to reduce the potential large amounts of data moved to the client (which, by default, is all data from all tables).

For more information on SQL Server replication, the Publication Wizard, and merge publications, see the SQL Server documentation.

A user takes the data offline

When a user is viewing a Page in Microsoft Internet Explorer or in Page view of Access, and takes the data offline:

- Internet Explorer will automatically download to the user's computer the Access project, the Page, and, if a CDF file is defined, all other Pages and HTML-related files defined.
- SQL Server does a merge synchronization that creates a snapshot of the data in the remote SQL Server database based on the named, merge publication and downloads this data snapshot to SQL Server 2000 Desktop Edition running on the user's computer.

A user works with the offline data

A user disconnects the computer from the intranet, and then reads and updates the data on the local computer. While working offline, the connection to the Access project is automatically pointed to the local subscription database instead of the remote server database.

A user brings the offline data back online

A user reconnects a client computer to the corporate intranet and synchronizes the Page or Pages. The Access project is automatically pointed to the remote server database, instead of the local subscription database. SQL Server merges and updates the offline data (including any schema changes) from the local computer to the remote SQL Server database. If necessary, SQL Server automatically resolves data conflicts with data changed while you were offline as defined by the merge publication rules.

Make a data access page available offline (ADP)

1. Open a data access page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.
2. On the **Favorites** menu, click **Add to Favorites**.
3. In the **Add Favorite** dialog box, select **Make available offline**.

Note If the **Make available offline** check box is unavailable, you won't be able to take a data access page offline. Contact the designer of the page for assistance.

4. Click **OK**.

The page and its associated data are saved to your computer.

Make an offline data access page online (ADP)

1. Open the data access page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.
2. On the **File** menu, click **Work Offline** so that it is no longer selected.
3. On the **Favorites** menu, right-click the entry for the data access page, and then click **Make Available Offline** so that it is no longer selected.
4. In the **Confirm Offline Item Delete** dialog box, click **Yes**.

Switch between offline and online modes of a page (ADP)

1. Open a data access page in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.
2. On the **File** menu, click **Work Offline**.

If **Work Offline** is selected, the page is offline. If it is not selected, the page is online.

Note A page will be offline if your computer is not connected to a network when you open the page.

Synchronize an offline data access page with source data (ADP)

1. Open Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.
2. On the **Tools** menu, click **Synchronize**.
3. In the **Items to Synchronize** dialog box, select the check box for the data access page you want to synchronize.
4. Click **Synchronize**.

The Microsoft Synchronization Manger merges the offline changes with the data in the source database.

Note The rules for merging offline data with the source database are defined either by the designer of the page or the database administrator. Contact them if you encounter errors or conflicts during the synchronization process. You might lose some of your changes during the merge operation, depending on how the rules are defined.

[Show All](#)

Show or update the link to a data access page

1. In the [Database window](#), click **Pages**  under **Objects**.
2. Right-click the page you want, and then click **Properties**.

The **Path** field shows the current path to the selected page.

3. To update the path to the page, either edit the contents of the **Path** field or click **Build** .

[Show All](#)

I get a message when I attempt to enter data in Page view.

When you try to move out of a field after changing data, Microsoft Access validates the data, making sure that the [value](#) you entered is allowed in that field. If the value isn't allowed, Access alerts you. In order to move off the field, you must either change the data to an acceptable value or undo your changes.

A value might not be allowed in a field for the following reasons:

- The value isn't compatible with the field's data type. If you enter a value that doesn't match the data type, Access attempts to convert the entry to the correct data type. However, some data can't be converted to other data types. For example, you can't store a name in a field with the Number data type in a [Microsoft Access database](#) or the smallint data type in a [Microsoft Access project](#).
- The data breaks a rule set in the **ValidationRule** property for the field.
- Data is required in the field, and you've tried to leave it blank.
- The validation macro specified for the **BeforeUpdate** property canceled the update.

Access also validates data when you move out of a record. Before it saves a record, Access makes sure that your changes don't break any [validation rules](#) set in the **ValidationRule** property for the record. Additionally, if the table you are editing has a [primary key](#) or [unique index](#), Access makes sure that the record you are editing doesn't have the same value for the fields as another record in the table. If the record can't be saved, you must either make any necessary corrections, or cancel your changes by clicking **Undo**  or pressing ESC.

[Show All](#)

I get the message "The DefaultSort property is invalid."

In the **DefaultSort** property box, type the name of the field or fields you want to sort on. Separate multiple field names with a comma. If you want to sort in ascending or descending order, you can follow each name with a space and the keyword **ASC** or **DESC**, respectively. For example, type **OrderDate DESC, OrderID ASC**. If you don't specify the order, Microsoft Access sorts in ascending order.

If the control you want to sort on is bound to an [expression](#), you must type the alias of the control as displayed in its **ControlSource** property. For example, if the **ControlSource** property of the control you want to sort on is set to **GroupOfExtendedPrice: ExtendedPrice**, type **GroupOfExtendedPrice** in the **DefaultSort** property box.

[Show All](#)

I get the message "Invalid argument."

- In the regional settings of Microsoft Windows Control Panel, your location is set to a region that uses a character other than a period (.) as a decimal separator. The settings for properties in data access pages require that the decimal separator be a period for all languages. Either change the region or set the decimal symbol to a period in Windows Control Panel regional settings.
- The [value](#) you typed in the property sheet does not match the [data type](#) of the property. For example, the **Width** property of a section can be set to any number, but if you type a string such as **Narrow**, you will get the "Invalid argument" message.

[Show All](#)

I get the message "The expression is not valid."

You attempted to type an [expression](#) that includes the name of a field that is not in the [data definition](#) of the page. Any field that you reference in an expression on a data access page must be in that page's data definition. To add a field to the page's data definition, drag the field from the [field list](#) to the data access page. If you don't want to show the [values](#) in the field on the page, you can hide or delete the control bound to the field. However, even if you delete the control, a field referenced in an expression remains part of the page's data definition.

[Show All](#)

Troubleshoot HTML formatted files

[My server-generated HTML files do not appear as expected when I display them in a Web browser.](#)

When you export a [datasheet](#) to [ASP](#) or [IDC/HTX](#) output format, the data that you see when you display the file in a Web browser is the actual data in the recordset behind the datasheet. By default, Microsoft Access saves Text and Memo fields with left alignment and saves all other supported data types with right alignment.

When you export a datasheet, Access does not output [OLE objects](#) or the related values from a Lookup field, and ignores the **OrderBy**, **Filter**, **Format**, and **InputMask** property settings.

If you want the server-generated HTML files to look like the datasheet in an Access database, you can create and export a query that has the same recordset as the datasheet. When you create the query, do the following:

- Use the **Sort** cell in the query design grid to sort the data.
- Use the **Criteria** cell in the query design grid to filter the data.
- Use the **Format** function to format the fields, such as Currency, Date/Time, and Yes/No.
- Create an Autolookup query to display the values from a Lookup list.

[I can't open a file on the Internet.](#)

The file may no longer be available. Check with the Web site administrator for the status of the file.

You may not have the correct permissions to access the file. Check with the Web site administrator to determine the permissions needed to access the file.

You must have access to the Internet— for example, you could have a modem and an Internet service provider, or you could have access through your corporation's network— to open files at an [FTP](#) site or on the [World Wide Web](#).

If you have Internet access and you still can't open the file, the site may be too busy. Try to open the file later.

 [I want to open an HTML file in Access, but it opens in another Office program](#)

In the **Open** dialog box (**File** menu), when you select an [Hypertext Markup Language](#) (HTML) file (also called a Web page) and click **Open** or double-click the file, the file opens in the program that created the file. For example, if you attempt to open an HTML format file that was created in Microsoft Word, the file opens in Word— not in Microsoft Access as a data access page.

1. To open an HTML file that was created in another Office program as a data access page in Access, do one of the following:

In Windows 2000:

- Right-click the file in the **Open** dialog box, point to **Open with**, and then click **Microsoft Access** on the shortcut menu.

Note If the file type that you are trying to open hasn't previously been associated with another program, instead of pointing to the **Open with** dialog, click the **Open with** dialog, and select **Microsoft Access** from the list of programs.

 [Supported Web page software](#)

The following table summarizes supported products, platforms, and Web page output format files.

Web page output file format	Supported product and platform
	Dynamic HTML 4.0 or later
	Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later
	All Web server products and platforms
Server-generated HTML, ASP	All Web browsers

(* .asp)

- IIS 5.0 on Windows 2000 Server

Server-generated HTML,
IDC/HTX (*.idc, *.htx)

All Web browsers

- IIS 5.0 on Windows 2000 Server

HTML 3.2 or later

Static HTML (*.htm, *.html)

All Web browsers

All Web server products and platforms

Note To use any Web page from a client computer, you need a Web browser, such as Microsoft Internet Explorer, and a connection to the Internet or intranet server where the Web pages are located. For information about how to set up Internet access, see Windows Help.

 [I get an error message when I try to import or link to data on a FTP or HTTP Server](#)

Microsoft Access does not support importing or linking data from a remote FTP or HTTP server. Save the file or HTML page to your hard drive or intranet site, and then import or link the file from there.

[Show All](#)

I get the message "The setting you entered isn't valid for this property."

If you are entering a [value](#) in a property that has a drop-down list, select a value from the list. For information about valid values for properties that don't have drop-down lists, press F1 in the property box.

If you are entering an [expression](#) in the **ControlSource** property box, make sure:

- You precede the expression with an equal sign (=) or an alias for the expression. For example, both of the following are valid settings:

`=Quantity*UnitPrice`

`ExtendedPrice: Quantity*UnitPrice`

- The data access page is bound to a table, [query](#), or other [record source](#) before you attempt to enter the expression. You can't use expressions on a data access page that isn't bound to data.
- The **ID** property of a [calculated control](#) is set to a unique name. Don't use the name of one of the fields you used in the expression or the ID of another control on the data access page.

[Show All](#)

I get the message "Undefined function in an expression."

You've bound a control on a data access page to a [calculated field](#) that uses a function that isn't supported on data access pages.

[Functions supported on a data access page](#)

A [control](#) in a [data access page](#) or a form that is open in [PivotTable view](#) or [PivotChart view](#) can contain an [expression](#) that uses a Microsoft Visual Basic for Applications (VBA) function. However, the expression will generate an error unless it appears on the following list.

Note This limitation doesn't apply to expressions in tables, queries, views, stored procedures, or user-defined functions that you have opened in PivotTable or PivotChart view.

[A - C](#)

ABS	ARRAY
ASC	ASCB
ASCW	ATN
CBOOL	CBYTE
CCUR	CDATE
CDBL	CHOOSE
CHR	CHR\$
CHRB	CHRB\$
CHRW	CHRW\$
CINT	CLNG
COS	CSNG
CSTR	CVAR
CVDATE	CVERR

[D - H](#)

DATE	DATE\$
DATEADD	DATEDIFF
DATEPART	DATESERIAL
DATEVALUE	DAY
DDDB	ERROR
ERROR\$	EXP
FIX	FORMAT
FORMAT\$	FV
HEX	HEX\$
HOUR	

[I - L](#)

IIF	IMESTATUS
INSTR	INT
IPMT	IRR
ISDATE	ISEMPTY
ISERROR	ISNULL
ISNUMERIC	ISOBJECT
ICASE	ICASE\$
LEFT	LEFT\$
LEFTB	LEFTB\$
LEN	LENB
LOG	LTRIM
LTRIM\$	

[M - Q](#)

MID	MID\$
MIDB	MIDB\$
MINUTE	MIRR
MONTH	NOW

NPER NPV
OCT OCT\$
PARTITION PMT
PPMT PV
QBCOLOR

[R- S](#)

RATE RGB
RIGHT RIGHT\$
RIGHTB RIGHTB\$
RND ROUND
RTRIM RTRIM\$
SECOND SGN
SIN SLN
SPACE SPACE\$
SQR STR
STR\$ STRCOMP
STRCONV STRING
STRING\$ SWITCH
SYD

[T- Z](#)

TAN TIME
TIMES\$ TIMER
TIMESERIAL TIMEVALUE
TRIM TRIM\$
TYPENAME UCASE
UCASE\$ VAL
VARTYPE WEEKDAY
YEAR

[Show All](#)

About copying a database object

You can easily customize an existing [database object](#) in a [Microsoft Access database](#) or [Microsoft Access project](#) by creating a copy of that object and modifying the copy. Also, you might want to make a backup copy of a database object. For example, you might want to make a backup copy before making design changes, or before running an append or update query that changes the data in a table.

A copy of the database object usually inherits the properties associated with the original database object. However, if you create a copy of a [table](#) in an Access project, the copy does not inherit the [constraints](#) and [triggers](#) of the original table. Also, if you create a copy of a table or query in an Access database, and you want a [subdatasheet](#) in the table or query to exist in the copy as well, you must open the copy and insert the subdatasheet.

When you create a copy of a [data access page](#), Access prompts you to provide a name for the new HTML file.

[Show All](#)

Copy a database object

[Copy a database object within a Microsoft Access file](#)

1. In the [Database window](#) under **Objects**, click the type of [database object](#) that you want to copy.
2. In the Object list on the right side of the Database window, click the object and then click **Copy**  on the toolbar.

If you are copying the object to a different [Microsoft Access database](#) or [Microsoft Access project](#), close the current Access database or Access project, and then open the Access database or Access project into which you want to paste the object.

3. Click **Paste**  on the toolbar.
4. Type a unique name for the object in the **Paste As** dialog box.

If you are in an Access database that is a [Design Master](#), and if you want the new object to be replicated to other databases, select the **Make Replicable** check box.

[Copy the structure of a table or append data to an existing table](#)

1. In the [Database window](#) under **Objects**, click **Tables**.
2. In the Object list on the right side of the Database window, click the table whose structure or data you want to copy, and then click **Copy**  on the toolbar.
3. Click **Paste**  on the toolbar.
4. Do one of the following:

To paste the structure of the table, click **Structure Only** under **Paste Options**.

To append data, in the **Table Name** box, type the name of the table you're appending data to, and then click **Append Data To Existing Table**.

[Copy a database object to another instance of Access or to another Microsoft application](#)

You can copy a table, query, or report to another Microsoft application that is running on your computer.

1. For each program except Microsoft Access and the application that you want to copy the object to, click the **Minimize** button in the upper-right corner of the window.
2. Right-click the Microsoft Windows task bar, and then click **Tile Windows Vertically**.
3. Drag the object from the [Database window](#) to the other application.

[Show All](#)

Delete a database object

1. Close the [database object](#) you want to delete. In a [multiuser](#) environment, confirm that all users have closed the database object.
2. Under **Objects** in the [Database window](#), click the type of database object you want to delete.
3. Click the object in the Object list, and then press DELETE.

Notes

- When you delete a [data access page](#) from the current database, Microsoft Access will ask you if you want to delete the link and the page, or just the link. Deleting just the link removes the page from the current database, but it doesn't remove the page HTML file from your computer.
- When you delete a table, query, or [database diagram](#) in a Microsoft Access project, you delete that object from the Microsoft SQL Server database that the Access project is connected to.

[Show All](#)

Rename a database object

1. Close the [database object](#) you want to rename. In a [multiuser](#) environment, confirm that all users have closed the database object.
2. Under **Objects** in the [Database window](#), click the object type of the object you want to rename.
3. Right-click the object, and then click **Rename**.
4. Type the new name for the object, and then press ENTER.

Note Renaming a [data access page](#) renames the link to the HTML file, but it does not rename the HTML file itself. Use My Computer or Microsoft Windows Explorer to rename the HTML file.

[Show All](#)

Preview a database object

To preview the [datasheet](#) of a table, query, or form, click **Print Preview**  on the **Table Datasheet**, **Query Datasheet**, or **Form View** toolbar.

 [Preview a form](#)

 [Preview an open form](#)

- Click **Print Preview**  on the toolbar.

The preview depends on the view in which the form is currently open, except when the form is open in Design view. If you preview a form in Design view, you see the form in Form view.

 [Preview a form from the Database window](#)

1. Click **Forms**  under **Objects**.
2. Click the form you want to preview.
3. Click **Print Preview**  on the toolbar.

You see the form in the view specified by the **DefaultView** property of the form.

 [Preview a report](#)

Previewing a report shows you how it will look on the printed page.

 [Preview the layout of a report](#)

In Layout Preview, you can take a quick look at the layout, which includes just a sample of the data in the report. It does not show the actual data in the report.

- In report [Design view](#), click the arrow next to the **View** button on the toolbar, and then click **Layout Preview**.

Note If the report is based on a [query](#) that prompts you for parameters, you don't need to enter any values; just click **OK**. Microsoft Access ignores the parameters when you choose **Layout Preview**.

[Preview the data in a report](#)

In Print Preview, you can verify the actual data by reviewing every page in the report.

[Preview a report from Design view](#)

- Click **Print Preview**  on the toolbar.

[Preview a report from the Database window](#)

1. Click **Reports**  under **Objects**.
2. Click the report you want to preview.
3. On the Database window toolbar, click the **Preview** button.

[Preview a data access page](#)

To preview a data access page, you must have Microsoft Internet Explorer 5.5 or later on your computer.

1. Open a [data access page](#) in [Design view](#).
2. Click the arrow next to the **View**  button and select **Web Page Preview**.
3. In Internet Explorer, click **Print Preview** on the **File** menu.

[Show All](#)

View a list of database objects

1. With the [Microsoft Access database](#) or [Microsoft Access project](#) open, click **Database Properties** on the **File** menu.
2. Click the **Contents** tab.

[Show All](#)

Save a database object

1. Click **Save**  on the **Design** toolbar.
2. Do one of the following:
 - If you are saving a [database object](#) for the first time, type a unique name in the **Save As** dialog box, and then click **OK**. If you click **Cancel** in the **Save As** dialog box, the object is not saved.
 - If you are saving a [data access page](#) for the first time, specify a location and name for the HTML file in the **Save As Data Access Page** dialog box.

To save the page so that you can use it over a network, select a network location in the **Save in** box, or specify a [UNC path](#) in the **File name** box. If you save the page to location that specifies an absolute path (a path that contains a drive letter), Microsoft Access displays a message stating that the page might not be able to connect to data through the network. To connect through a network, edit the connection string to specify a network (UNC) path.

Notes

- You don't need to save new data in [records](#). Access automatically saves a record when you do any of the following: move the [focus](#) to a different record; close the active form, datasheet, data access page, or database; or quit Access.
- Access automatically saves changes to your data whenever you quit. However, if you have made changes to the design of any database objects since you last saved them, Access asks whether you want to save these changes before quitting.
- If you make a change in [PivotTable view](#), Access updates the [PivotChart view](#) automatically. Similarly, if you make a change in PivotChart view, Access automatically update the PivotTable view. When you save the object, you also save the PivotTable and PivotChart views.

[Show All](#)

Save a database object as another object type

If the [database object](#) is open, you must save your changes before you save it as another object.

1. In the [Database window](#), under **Objects**, click the type of object that you want to convert.
2. In the **Object** list, click the object that you want to save as another object type, and then click **Save As** on the **File** menu.
3. In the **Save As** dialog box, type a unique name in the **Save <object name> To** box, and then click the object type that you want.
4. If you are in a [Microsoft Access database](#) that is a [Design Master](#), and if you want the new object to be replicated to other databases, select the **Make Replicable** check box.
5. Click **OK**.
6. If you chose to save the object as a data access page, in the **New Data Access Page** dialog box, type a name for the page. If a default page folder is specified in the **Options** dialog box (**Tools** menu), Microsoft Access will prompt you to save the page in the default location.

Access creates the database object and opens it in an appropriate view. If the new object is a page, Access also creates a shortcut for the page in the current database.

Note You cannot save a [data access page](#) or a [module](#) as another object type.

[Show All](#)

Show or change the general properties of a database object

1. Under **Objects** in the [Database window](#), click the type of [database object](#) whose properties you want to change.
2. Click the database object.
3. Click **Properties**  on the toolbar.

[Show All](#)

Set or change the description of a database object

The content of the **Description** field in the **Properties** dialog box is what is displayed under **Description** in the [Database window](#) when you click **Details**  on the toolbar.

1. Under **Objects** in the Database window, click the type of [database object](#) whose description you want to set or change.
2. Click the database object.
3. Click **Properties**  on the toolbar.
4. Type the new description in the **Description** box.

[Show All](#)

Show or hide shortcuts for creating new database objects

When you click **Objects** in the [Database window](#) and then click an object type, the Object list on the right side of the Database window contains shortcuts that you can use to create a new [database object](#) of that type.

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Select or clear the **New object shortcuts** check box.

[Show All](#)

Show or hide database objects

[Specify whether a database object is hidden or visible](#)

1. Under **Objects** in the [Database window](#), click the type of database object whose properties you want to change.
2. Click **Properties**  on the toolbar.
3. Select or clear the **Hidden** check box.

Note In a [Microsoft Access project](#), you cannot change the properties of a table, query, or [database diagram](#) because these objects reside in the Microsoft SQL Server database. However, you can change the properties of a form, report, macro, or module, because these objects reside in the Access project itself, not in the Microsoft SQL Server database. You can also change the properties of a [data access page](#).

[Show or hide objects defined as hidden by default](#)

If you need to work with objects that are defined as hidden without changing their status as hidden, you can display them in the [Database window](#).

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **Hidden objects** check box. Microsoft Access displays hidden objects with dimmed icons to distinguish them from objects that aren't defined as hidden.

[Show or hide system objects by default](#)

Microsoft Access automatically creates [system objects](#) when you create a new database.

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **System objects** check box.

[Show All](#)

About groups of database objects

You can organize different types of [database objects](#) into a group. For example, you can manage a form with multiple subforms more easily if you create a group that contains the main form, the subforms, and the tables or queries that they are based on. You can also create a group for the forms and reports that you use most often; when you click the icon for that group, both the forms and the reports appear in the Object list in the [Database window](#).

A group consists of shortcuts to the database objects that belong to it; adding an object to a group does not change that object's original location. When you click **Objects** and then click the icon for a particular object type, any database object of that type appears in the Object list— even if that object also belongs to a group. When you click the **Groups** bar in the Database window and then click the icon for your group, all of the database objects that you have added to the group appear in the Object list.

Deleting a database object shortcut from a group does not delete the object itself. For example, if you delete a form shortcut from your group, and then click **Forms**  under **Objects** in the Database window, that form still appears in the list of all forms.

[Show All](#)

Group database objects

1. Press F11 to switch to the [Database window](#).
2. If groups aren't displayed, click the **Groups** bar in the column on the left side of the Database window.
3. Right-click anywhere under the **Groups** bar, and then click **New Group** on the shortcut menu.
4. Type a name for your group in the **New Group Name** box.

[Show All](#)

Add a database object to a group

1. Press F11 to switch to the [Database window](#).
2. Drag the object or shortcut from the Object list on the right side of the Database window to the icon for the group you want to add it to. This icon is located under the **Groups** bar in the column on the left side of the Database window.

Note Adding an object to a group creates a shortcut to the object in that group, but it doesn't affect the object's actual location. For example, if you add a form to your group, and then click **Forms**  under **Objects** in the Database window, that form still appears in the list of all forms.

[Show All](#)

Remove a database object from a group

1. Press F11 to switch to the [Database window](#).
2. Click the **Groups** bar in the column on the left side of the Database window, and then click the group that contains the database object shortcut you want to delete.
3. Right-click the shortcut, and then click **Delete**  on the shortcut menu.

Note Removing a database object shortcut from a group only deletes the shortcut to that object; it does not delete the object itself.

[Show All](#)

Rename a database object in a group

1. Press F11 to switch to the [Database window](#).
2. Click the **Groups** bar in the column on the left side of the Database window, and then click the group that contains the database object shortcut you want to rename.
3. Right-click the shortcut.
4. Click **Rename** on the shortcut menu, and then type a new name for the shortcut.

Note Renaming a database object shortcut in a group changes only the name of the shortcut; it does not change the name of the object itself.

[Show All](#)

Delete or rename a group of database objects

1. Press F11 to switch to the [Database window](#).
2. If groups are not displayed, click the **Groups** bar in the column on the left side of the Database window.
3. Right-click the group you want to delete or rename, and then do one of the following:

To delete the group, click **Delete Group** on the shortcut menu.

To rename the group, click **Rename Group** on the shortcut menu, and then type a new name for the group in the **New Group Name** box.

[Show All](#)

About types of controls in Access

Microsoft Access includes the following types of controls, which are all accessible through the [toolbox](#) in [Design view](#) of a form, report, or data access page: text box, label, option group, option button, check box, list box, command button, tab control, image control, line, rectangle, and ActiveX custom controls. You can also add a Microsoft Office PivotTable list, an Office Chart, or an Office Spreadsheet to a form, report, or [data access page](#).

Forms and reports have these additional controls: toggle button, combo box, bound object frame, unbound object frame, and page break. You can also add a subform or subreport to a form or report. Data access pages also include the drop-down list box, hyperlinks, scrolling text, and the PivotTable list, spreadsheet, and chart components.

Controls are objects on a form, report, or data access page that display data, perform actions, or are used for decoration. For example, you can use a text box on a form, report, or data access page to display data, a command button on a form to open another form or report, or a line or rectangle to separate and group controls to make them more readable.

All the information on a form or report is contained in controls. On data access pages, information is contained in controls in the same way as it is on forms and reports. However, information can also be typed directly on the data access page.

Controls that you can use to display, enter, filter, or organize data in Access



You use text boxes on a form, report, or [data access page](#) to display data from a [record source](#). This type of text box is called a bound text box because it's bound to data in a field. Text boxes can also be unbound. For example, you can create an unbound text box to display the results of a calculation or to accept input from a user. Data in an unbound text box isn't stored anywhere.

Product Name:	ProductName
Unit Price:	UnitPrice
Emp. Price:	=[UnitPrice]*0.75

- 1 These bound text boxes display data from the ProductName and UnitPrice fields in the Products table.
- 2 This unbound text box displays the result of a calculation.

[Labels](#)

You use labels on a form, report, or [data access page](#) to display descriptive text such as titles, captions, or brief instructions. Labels don't display values from [fields](#) or [expressions](#); they're always [unbound](#) and they don't change as you move from record to record.

A label can be attached to another [control](#). When you create a text box, for example, it has an attached label that displays a caption for that text box. This label appears as a column heading in the [Datasheet view](#) of a form. When you create a label by using the **Label** tool , the label stands on its own— it isn't attached to any other control. You use stand-alone labels for information such as the title of a form, report, or data access page, or for other descriptive text. Stand-alone labels don't appear in Datasheet view.

1

To look at products for this supplier: **Review Products**

Contact Name: Charlotte Cooper

Co. Name: Exotic Liquids

2

- 1 Stand-alone label
- 2 Labels attached to text boxes

[List boxes, combo boxes and drop-down list boxes](#)

In many cases, it's quicker and easier to select a value from a list than to

remember a value to type. A list of choices also helps to ensure that the value that's entered in a field is correct. If you have enough room on your form or [data access page](#) to display the list at all times, you might want to use a list box. When you want to create a control that displays a list but requires less room, use a [combo box](#) if you are working in a form, or use a [drop-down list box](#) if you are working in a data access page.

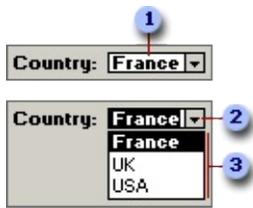
List boxes The list in a list box consists of rows of data. In a form, a list box can have one or more columns, which can appear with or without headings. If a multiple-column list box is [bound](#), Access stores the values from one of the columns. In a data access page, a list box has one column without a heading.



- 1 When you click a value in a list box...
- 2 ...if the list box is bound, Access inserts the value selected into the field that the list box is bound to. If the [bound column](#) is different from the column displayed in the list, Access inserts the value from the bound column instead of the value you clicked.
- 3 This is the underlying record source that the form is bound to.

You can use an [unbound](#) list box to store a value that you can use with another [control](#). For example, you could use an unbound list box to limit the values in another list box or in a [custom dialog box](#). You could also use an unbound list box to find a record based on the value you select in the list box.

Combo boxes A combo box is like a text box and a list box combined, so it requires less room. You can type new values in it, as well as select values from a list. The list in a combo box consists of rows of data. Rows can have one or more columns, which can appear with or without headings.



- 1 Type text directly into a combo box...
- 2 ...or click the arrow...
- 3 ...to see a list of values to choose from.

When you enter text or select a value in a [bound](#) combo box, the entered or selected value is inserted into the field that the combo box is bound to. If a multiple-column combo box is bound, Access stores only the value from the bound column. You can use an [unbound](#) combo box to store a value that you can use with another [control](#). For example, you could use an unbound combo box to limit the values in another combo box or in a [custom dialog box](#). You could also use an unbound combo box to find a record based on the value you select in the combo box.

Drop-down list boxes On a data access page, you can use a [drop-down list box](#) instead of a list box. A drop-down list box on a data access page looks like a combo box on a form. As in a combo box, a drop-down list box shows only one record until you click to expand the contents; however, you can't type new values in a drop-down list box. The list in a drop-down list box consists of rows of data. Rows can have only one column that appears without headings.



- 1 Click the arrow...
- 2 ...to see a list of values to choose from.

When you select a value in a drop-down list box that is bound to a field, the selected value is inserted into that field. You can also use an unbound drop-down list box to store a value that you can use with another control. For example, you could use an unbound drop-down list box to limit the values in another drop-down list box. You could also use an unbound drop-down list box to find a

record based on the value you select in the drop-down list box.

[Command buttons](#)

Command buttons provide you with a way of performing action(s) by simply clicking them. When you choose the button, it not only carries out the appropriate action, it also looks as if it's being pushed in and released.

[What are command buttons?](#)

You use a command button on a form or [data access page](#) to start an action or a set of actions. For example, you can create a command button that opens another form. To make a command button do something on a form, you write a [macro](#) or [event procedure](#) and attach it to the button's **OnClick** property. On a data access page, you can attach code written in either Microsoft JScript or Microsoft [Visual Basic Scripting Edition \(VBScript\)](#) to a command button by using the [Microsoft Script Editor](#).

You can display text or a picture on a command button in a form; you can display only text on a command button in a data access page.

[Creating command buttons](#)

You can create a command button on your own, or you can have Microsoft Access create your command button for you by using a wizard. A wizard speeds up the process of creating a command button because it does all the basic work for you. When you use a wizard, Access prompts you for information and then creates the command button based on your answers. By using the wizard, you can create more than 30 different types of command buttons. You can create command buttons to:

- Dial a phone number.
- Run a query or macro.
- Run or quit an application.
- Edit or apply a filter.
- Print or mail a report.
- Print the current record.
- Update form data.
- Find a specific record.

It's also a good idea to use the Command Button Wizard if you want to learn how to write event procedures. When Access creates a command button in a form or report with a wizard, it creates an event procedure and attaches it to the button. You can open the event procedure to see how it works and modify it to fit your needs. When you create a command button on a data access page with a wizard, Access doesn't create event procedures for you.

Wizards are not available in a stand-alone data access page or when you point a data access page to a database other than the one currently open. The Command Button Wizard is also not available if a data access page isn't bound to a table or query. You will have to create a command button on your own if wizards aren't available.

[Check boxes](#)

You can use a check box on a form, report, or [data access page](#) as a stand-alone [control](#) to display a **Yes/No** value from an underlying table, query, or SQL statement. For example, the check box in the following illustration is [bound](#) to the Discontinued field in the Products table. The data type of the Discontinued field is **Yes/No**. If the box contains a check mark, the value is **Yes**; if it doesn't, the value is **No**.



1 This stand-alone check box is bound to the Discontinued field in the Products table.

When you select or clear a check box that's bound to a **Yes/No** field in a [Microsoft Access database](#) or a **Bit** column in an [Access project](#), Access displays the value in the underlying table according to the field's **Format** property (**Yes/No**, **True/False**, or **On/Off**). In an Access project, the option button is bound to a column defined as a **Bit** data type. When the value in this column is 1, it is equivalent to **Yes**, **On**, or **True** conditions. When the value is 0, the column indicates **No**, **Off**, or **False** conditions.

You can use check boxes in an [option group](#) to display values to choose from.

[Option buttons](#)

You can use an option button on a form, report, or [data access page](#) as a stand-alone [control](#) to display a **Yes/No** value from an underlying [record source](#). For example, the option button in the following illustration is [bound](#) to the Discontinued field in the Products table of a database. The data type of the Discontinued field is **Yes/No**. If the option button is selected, the value is **Yes**; if not, the value is **No**.



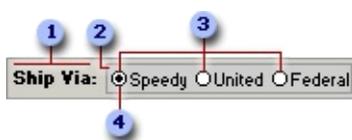
1 This stand-alone option button is bound to the Discontinued field in the Products table. The value in the field is **Yes**.

When you select or clear an option button that's bound to a **Yes/No** field in a [Microsoft Access database](#), the value in the underlying table displays according to the field's **Format** property (**Yes/No**, **True/False**, or **On/Off**). In a [Microsoft Access project](#), the option button is bound to a column defined as a **Bit** data type. When the value in this column is 1, it is equivalent to **Yes**, **On**, or **True** conditions. When the value is 0, the column indicates **No**, **Off**, or **False** conditions. You can also use option buttons in an [option group](#) to display values to choose from.

[Option groups](#)

You can use an option group on a form, report, or [data access page](#) to display a limited set of alternatives. An option group makes selecting a value easy because you can just click the value that you want. Only one option in an option group can be selected at a time. If you want to present more than a few options, use a [list box](#), a [combo box](#), or a [drop-down list box](#) instead of an option group.

In a form or report, an option group consists of a group frame as well as a set of check boxes, option buttons, or toggle buttons. In a data access page, an option group consists of a group frame and a set of option buttons.



- 1 Label
- 2 The option group is the frame that surrounds the controls inside it.
- 3 Option buttons
- 4 Only one option in an option group can be selected at a time.

If an option group is bound to a field, only the group frame itself is bound— not the check boxes, toggle buttons, or option buttons inside the frame. Because the **ControlSource** property of the group frame is set to the field that the option group is bound to, you don't set the **ControlSource** property for each [control](#) in the option group. Instead, you set the **OptionValue** (form or report) or the **Value** (data access page) property of each check box, toggle button, or option button. In a form or report, set the control property to a number that's meaningful for the field the group frame is bound to. In a data access page, set the control property to either a number or any text that's meaningful for the field the group frame is bound to. When you select an option in an option group, Access sets the value of the field to which the option group is bound to the value of the selected option's **OptionValue** or **Value** property.



- 1 **OptionValue** property set to 1; when you click this button, Access sets the value of the Ship Via option group to 1 and stores this number in the underlying table.
- 2 **OptionValue** property set to 2
- 3 **OptionValue** property set to 3

Notes

- The **OptionValue** or **Value** property is set to a number because the value of an option group can only be a number, not text. Access stores this number in the underlying table. In this example, if you want to display the name of the shipper instead of a number in the Orders table in an [Access database](#), you can create a separate table called Shippers that stores shipper names, and then make the ShipVia field in the Orders table a Lookup field that

looks up data in the Shippers table.

- In a form or report, an option group can also be set to an [expression](#).

[Toggle buttons](#)

You can use a toggle button on a form as a stand-alone [control](#) to display a **Yes/No** value from an underlying [record source](#). For example, the toggle button in the following illustration is [bound](#) to the Discontinued field in the Products table of a database. The data type of the Discontinued field is **Yes/No**. When the button is pressed in, the value in the Products table is **Yes**. When the button isn't pressed in, the value in the Products table is **No**.



1 This stand-alone toggle button is bound to the Discontinued field in the Products table. The button is pressed in, so the value in the field is **Yes**.

When you click a toggle button that's bound to a **Yes/No** field in a [Microsoft Access database](#), the value in the underlying table displays according to the field's **Format** property (**Yes/No**, **True/False**, or **On/Off**). In an [Access project](#), the option button is bound to a column defined as a **Bit** data type. When the value in this column is 1, it is equivalent to **Yes**, **On** or **True** conditions. When the value is 0, the column indicates **No**, **Off**, or **False** conditions.

Toggle buttons are most useful when used in an [option group](#) with other buttons. In an option group, you can easily tell whether a button is pressed in.

Also, you can use pictures on toggle buttons instead of text. For example, instead of displaying the word "Discontinued" on the toggle button in the first example, you could display a picture of a trash can.

[Tabbed pages on forms](#)

You can use a tab control to present several pages of information as a single set. This is especially useful when you're working with many controls that can be sorted into two or more categories. For example, you might use a tab control on an Employees form to separate employment history and personal information.



- 1 Information about employment history is displayed on this page.
- 2 Personal information, such as home address and phone number, is displayed on this page.

Additional controls that you can use on data access pages



You can store HTML code in a field in a table, and when the [data access page](#) displays the values in that field, the HTML code performs the actions that you specified. You can accomplish this by binding the [bound span control](#) to a Text or Memo field in a [Microsoft Access database](#) or a text, ntext, varchar, or any other column that can store text in a [Microsoft Access project](#). The contents of a bound span control are not editable.

For example, if a value in a field is `red Text` and you set the **DataFormatAs** property to **HTML**, you will see red text when you view the data access page in [Page view](#). If the **DataFormatAs** property is set to **Text**, then `red Text` is the value displayed in the control. The default value of the **DataFormatAs** property is **Text**.

If the HTML text is lengthy, bind the bound span control to a field with the Memo data type (Access database), or the text or ntext data type (Access project).

Notes

- You can make the bound span control the default control for a grouped data access page. Set the **DefaultControlType** property of the data access page to **Bound Span**.
- You can use a bound span control to display calculations and data on a grouped data access page. In Microsoft Internet Explorer 5.01 with Service

Pack 2 (SP2) or later, bound span controls load faster than other controls, and hence improve the performance of the page.

- The bound HTML control in Access 2000 has been replaced by the bound span control in Access 2002 and later. Bound HTML controls on pages created using Access 2000 will automatically convert to bound span controls when you open the page in Access 2002 or later.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Hyperlink control](#)

Use the hyperlink control when each record in the underlying [record source](#) has a different hyperlink address. You bind the hyperlink control to a Text field that contains hyperlink values, such as e-mail addresses.

As you move from record to record in [Page view](#) or Microsoft Internet Explorer, you can click the hyperlink to either go to a different Web page or open a mail program to send an e-mail message.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Image hyperlink control](#)

Use the image hyperlink control to add an image to a data access page that, when clicked, displays another Web page from your hard drive, the Web, or another location. When you create an image hyperlink control, you select the image to display on the page and the address of the file to jump to. In [Page view](#), as the pointer passes over the image, the pointer changes to a hand, indicating that the image is a link that you can click to go to another page. You can also define [ScreenTips](#) and alternate text for the image.

The image hyperlink control is useful for providing links to information that relates to data on your data access page.

 [Scrolling text control](#)

Use a [scrolling text control](#), often called a [marquee](#), on a data access page to display moving text or text that scrolls. Scrolling text draws attention to a line of text, such as a headline or an important announcement. You display text that is contained in a field of your database by binding the scrolling text control to the field. You can customize scrolling text by setting options such as direction of travel, speed, and type of motion.

The text does not scroll in [Design view](#). To see the scrolling text, you have to open the page that contains the control in [Page view](#) or in Microsoft Internet Explorer.

[Show All](#)

Add a field to a form, report, or data access page

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Display the [field list](#) by clicking **Field List**  on the toolbar.
3. To select the field or fields that will serve as the basis for the [control](#), do one of the following:

To select	Do this
One field	Click the field.
A block of fields	Click the first field in the block, hold down the SHIFT key, and then click the last field in the block.
Nonadjacent fields	Hold down the CTRL key and click the name of each field that you want to include.
All fields (form or report only)	Double-click the field list title bar.
A table or query (data access page only)	Click the Wizards tool  in the toolbox if it's not already selected , and then click the table or query.



In a data access page, to quickly create one control that is based on a field, double-click that field in the field list.

4. Do one of the following:
 - Drag the field or fields from the field list to the form, report, or data access page.
 - Drag the table or query from the field list to the data access page. In the **Layout Wizard** dialog box, click the layout that you want to use.

Note You must drag the field or fields, the table, or the query from the field list. If you click the item in the field list and then click the form, report, or data access page, Microsoft Access won't create the control or controls. Access places one text box on the form, report, or data access page for each

field that you select in the field list. If you select a table or query in the data access page field list and then select **Columnar** or **Tabular** in the **Layout Wizard**, Access places a text box for each field in the table or query on the data access page. Each text box is bound to one field in the underlying data source.

5. Place the upper-left corner of the icon where you want the upper-left corner of the main control (not its label) to be, and release the mouse button.

Note Access creates the appropriate control for the field and sets certain properties, based on the field's data type. To specify the type of control you want to use, click the appropriate control tool in the toolbox before dragging the field from the field list.

[Show All](#)

Troubleshoot controls

[Troubleshoot expressions, data display and formatting, and the field list](#)

[#Error? or #Name? appears in a control.](#)

#Error? or #Name? may appear in a [control](#) for a number of reasons. To correct the problem, do the following:

- Make sure that the field specified in the control's **ControlSource** property hasn't been removed from the underlying table or other [record source](#).
- Check the spelling of the field name in the control's **ControlSource** property.
- If you specified an [expression](#) in the control's **ControlSource** property, make sure that there is an equal sign preceding the expression.
- Make sure that there are brackets around references in expressions to control or field names that include spaces. For example, to subtract a Shipped Date field from a Required Date field, enter the following expression: **=`[Required Date]-[Shipped Date]`**.
- If you are using one of the built-in functions, make sure that you're using the right number of arguments, that the arguments are in the right order, and that you haven't left out any necessary punctuation. To determine the correct syntax to use, see the Help topic for the function you're using.
- If you are referring to another control, make sure that you're using the correct syntax.
- Make sure that there isn't a circular reference to a control. For example, if you specify MyControl in the **Name** property of a control, and then type **=`[A]+[B]+[MyControl]`** in the **ControlSource** property box for the control, Microsoft Access can't process the expression.
- Make sure that the [Access file](#) doesn't contain a missing reference to a project or library by opening a [module](#) in [Design view](#), and clicking **References** on the **Tools** menu. Clear the check box next to any missing references, and then set the references you want.
- If the control is in a data access page and contains an expression that uses a function, make sure that the function is available in a data access page.

[I'm seeing unexpected results when I try to refer to a column in a list box or combo box in an expression.](#)

You use the **Column** property to refer to a column in a list box or combo box. The **Column** property might produce unexpected results for any of the following reasons:

- You haven't supplied an index number to the **Column** property. The index number identifies the column you're referring to. For example, Column(0) refers to the first column in a list box or combo box.
- You've specified an index number for the wrong column. Remember that index numbers for the **Column** property start at zero. Column(0) refers to the first column, Column(1) refers to the second column, and so on.
- You're performing a numeric calculation directly on the value returned by the **Column** property. Because the **Column** property returns a text value, you must convert it to a numeric value before performing a numeric calculation. For example, the following expression uses the **CCur** function to convert the value returned by the **Column** property to a currency value before multiplying it by the value of the Quantity field:

```
[Quantity] * CCur([PriceListBox].Column(1))
```

[When I try to create a bound control, the Field List isn't available.](#)

If you are working in a form or report, you might need to bind it to a [record source](#).

If you are working in a data access page, you might need to connect it to a database.

[I can't use text that contains an ampersand as a label or as a caption for a command button.](#)

Microsoft Access uses a single ampersand in a label or button to define an access key. Therefore, if you want to use an ampersand (&) in a label on a form or report, you must type two ampersands. For example, if you want the text "Products & Suppliers" to appear in a label, type **Products && Suppliers**.

[When I bind a control to a field, the control doesn't inherit some of the properties from the field](#)

If you bind a control to a field, or if you change the underlying field for a bound control, Microsoft Access doesn't set the property settings for the control to the corresponding property settings for the underlying field. In forms and reports in an [Access database](#), **ValidationRule**, **ValidationText**, and **DefaultValue** property settings in the underlying field will be applied, however, because these properties are enforced for the field, not just the control.

[I chose a raised or sunken appearance for a control in a form, but the three dimensional effect doesn't appear.](#)

Double-click **Display** in Microsoft Windows Control Panel, click the **Appearance** tab, and then click **Windows Standard** in the **Scheme** box.

[I can't select multiple controls at the same time in a data access page.](#)

To select multiple controls in a data access page, you must have Microsoft Internet Explorer 5.5 or later installed.

[Troubleshoot a control's property sheet](#)

[Why is the property sheet blank?](#)

If the property sheet is displayed and you click a [control](#) that's already selected, the property sheet will go blank. If you don't need to type a value in the control, press ESC to redisplay the property sheet. Otherwise, type a value and press ENTER.

[Why is the property box for a particular property blank?](#)

If the property setting for a particular property is the same for all selected [controls](#), that setting appears in the property sheet; otherwise, the property box for that property is blank. If you have multiple controls selected, and you change a setting for a property in the property box, that change affects all selected controls.

[Why don't I see all the properties in the property sheet?](#)

If you select multiple [controls](#) but don't see the properties you expect in the property sheet, you may have selected a control that you didn't mean to select. For example, you may have included a line in a group of text boxes. Properties appear in the property sheet only if they apply to all of the selected controls. To remove a control from a group selection, hold down the SHIFT key and click the control.

[Why don't the property settings for my control match the settings in table Design view?](#)

The property settings for a [bound control](#) might not match corresponding settings in the field in the underlying table that the [control](#) is bound to. If the settings are different, the form or report settings typically override those in the table, but they apply only to the current form or report. It's generally a good idea to set the **Format**, **DecimalPlaces**, **InputMask**, **ValidationRule**, **ValidationText**, and **DefaultValue** properties in the underlying table or query rather than in the control. This way, you can be sure that you have consistent settings for the field whenever you add it to a form or report.

[Troubleshoot a group of controls](#)

[I can't group text boxes or other controls.](#)

You must select more than one text box or other control at a time in order to create a control group.

- Select two or more controls by dragging a rectangle over the controls or by holding down the SHIFT key while you click each control. Then on the **Format** menu, click **Group**.

[I can't ungroup selected text boxes or other controls.](#)

You can ungroup selected text boxes or other controls only when one group at a time is selected on the form or report.

- Click any control in the group to select the group, and then on the **Format**

menu, click **Ungroup**.

[Troubleshoot conditional formatting in a text box or other control on a form or report](#)

[The conditional formatting I applied to a control doesn't change when I thought it would.](#)

If you have applied [conditional formatting](#) to a [control](#), then Microsoft Access applies the formatting depending on whether the control value or other evaluated data meets a condition that you specified.

- If you base the formatting condition on the text contained in the selected control by choosing **Field Value Is** in the **Conditional Formatting** dialog, then formatting such as bold or font color may change as the text in the control changes.
- Conditional formatting has priority over formatting that you apply directly to a control by using a **Formatting** toolbar button. If a specified condition is true, the formatting for that condition overrides the default formatting.

[Access isn't applying conditional formatting correctly to a control on my form or report.](#)

- **Check for multiple conditions** If you specify multiple conditions and a value meets more than one condition, Microsoft Access applies only the formatting for the first true condition.
- **See whether conditions overlap** If you specify conditions that overlap, Access applies only the format of the first true condition. For example, if you specify a condition that applies a yellow background to control values between 100 and 200 inclusive, and then you specify a second condition that applies a red background to control values below 120, the control values of 100 through 119 will have a yellow background. Try to avoid overlapping conditions.
- **Check control references** If you used an expression as the formatting criteria, the control references in the formula may not be correct.

[Troubleshoot check boxes, toggle buttons, option buttons, and option groups](#)

[I added a check box, toggle button, or option button to an option group, but the option group doesn't work.](#)

You might have placed the [control](#) outside the option group. When you add a check box, toggle button, or option button to an option group, Microsoft Access highlights the option group to show that the control will become part of the option group. If the option group isn't highlighted, you're probably placing the control too close to the edge of the option group.

[The option group on my data access page doesn't show or save data correctly.](#)

Make sure that the controls in your option group are option buttons. You can use check boxes in an option group on a form, but not in an option group on a data access page.

[Troubleshoot list boxes, combo boxes, and drop-down list boxes](#)

[The list box, drop-down list box, or combo box isn't sorted the way I want.](#)

If you use a wizard to create the list box, drop-down list box, or combo box, Microsoft Access automatically sorts the rows that make up the list by the first visible column. If you want to specify a different sort order for a control on a form or report, click the **Build** button next to the **RowSource** property box for the control, and specify a sort order in the Query Builder or SQL Statement Builder. If you want to specify a different sort order for a control on a data access page, create a query with the sort order you want, and then specify that query in the **ListRowSource** property of the control.

[The text displayed in the combo box is from the wrong column. How do I display data from another column?](#)

In a multiple-column combo box on a form or report, the text box portion of the combo box displays data from the first visible column in the list. You use the **ColumnWidths** property to determine which columns are visible. For example, if you have a combo box that is based on a query that includes the SupplierID field and the SupplierName field, in that order, then Microsoft Access displays the SupplierID field in the text box portion of the combo box unless you hide it

using the **ColumnWidths** property.

[When I select a value in a list, it disappears or gives me a message.](#)

You may have specified the wrong column in the **BoundColumn** (forms and reports) or the **ListBoundField** (data access pages) property box. The **BoundColumn** or **ListBoundField** property determines which column's data is stored in the field that is specified in the **ControlSource** property box. For example, if you have a combo box on a form or report that has a SupplierID column and a SupplierName column, the bound column would be the SupplierID column (the [foreign key](#) field) even if the SupplierName column is the only visible column.

[I can't select a value in the list box, drop-down list box, or combo box.](#)

You may have misspelled the field name in the **ControlSource** property box. You use the **RowSource** (forms and reports) or the **ListRowSource** (data access pages) property to specify the items that make up the list. When you click an item in the list, Microsoft Access needs to know where to store the value that you selected. You use the **ControlSource** property to specify where the values are stored.

[#Error? appears in the combo box or drop-down list box, or the combo box or drop-down list box is blank.](#)

In form [Design view](#), you probably entered an inappropriate setting in one of the following properties: **RowSource**, **ControlSource**, **BoundColumn**, **ColumnWidths**, **ColumnCount**. To learn more about combo box properties, see Microsoft Access Help, or click the property box for the property and press F1.

[I want to refer to a column in the list box or combo box other than the bound column.](#)

In a form, to refer to a column other than the bound column, use the **Column** property. The **Column** property is zero-based. For example, to refer to the second column in the combo box called SuppliersCombo on the Products form, use this syntax:

Forms![Products]![SuppliersCombo].Column(1)

[I can't type a new value in the combo box on a form.](#)

The **LimitToList** property is set to **Yes**. Set the **LimitToList** property to **No** so that you can type new values in the combo box. If the column displayed in the text box portion of the combo box isn't the bound column, Microsoft Access automatically sets the **LimitToList** property to **Yes**.

[Why isn't a value that I type in the combo box on a form added to the list?](#)

To automatically add a value to the list on a form, you need to write an [event procedure](#) and attach it to the combo box's **OnNotInList** property.

Note You must set the **LimitToList** property to **Yes** to use the NotInList event.

[My list box, drop-down list box, or combo box runs too slow. What can I do to speed it up?](#)

There are several things you can do to make your list boxes, combo boxes, and [drop-down list boxes](#) run faster:

- Base the list box, combo box, or drop-down list box on a saved [query](#) instead of an SQL statement. On a form, if you use a wizard to create a list box or combo box, Microsoft Access automatically sets the **RowSource** property of the control to an SQL statement. To change the **RowSource** property to a saved query, click the **Build** button next to the **RowSource** property box. With the **Query Builder** window open, click **Save** on the **File** menu, and type a name for the query. When you close the Query Builder window, click **Yes** when Microsoft Access asks if you want to update the property.
- Base the drop-down list box on a saved query instead of a table. On a data access page, if you use a wizard to create a drop-down list box, Access automatically sets the **ListRowSource** property of the control to the table you specified in the wizard. To retrieve rows from a query instead of a table, select the query name from the **ListRowSource** property box.
- In the query specified in the **RowSource** or **ListRowSource** property box, include only fields that are absolutely necessary. Extra fields can decrease performance.

- [Index](#) both the first field displayed in the combo box, list box, or drop-down list box and the bound field (if the fields are different).
- In combo boxes on a form, set the **AutoExpand** property to **No** if you don't need the fill-in-as-you-type feature.
- On a form, if the **AutoExpand** property for a combo box is set to **Yes**, the first displayed field should have a **Text** data type instead of a Number data type. In order to find a match in the list, Microsoft Access converts a numeric value to text. If the data type is **Text**, Microsoft Access doesn't have to do this conversion.
- On a form, if the bound field in a lookup combo box is not the displayed field:
 - Don't use [expressions](#) for the bound field or the displayed field.
 - Don't use restrictions in the row source.
 - Use single-table (or query) row sources rather than multiple-table row sources, if possible.
- Don't create list boxes, combo boxes or drop-down list boxes based on data in a linked table if the data won't change. It's better to import the data into your database in this case.

Note The Query Builder is not available in a data access page. Wizards are not available in a stand-alone data access page, or when you point a data access page to a database other than the one currently open. In both cases, you will have to create a list box or drop-down list box on your own.

 [Troubleshoot pictures and other OLE objects](#)

 [When I double-click a linked or embedded object, I get a message saying the file can't be opened.](#)

Microsoft Access displays this message when it can't open the source file for an OLE object. Check to make sure that:

- The application needed to edit the file is installed on your computer.
- You have enough memory to run the source application. Close other applications to free up memory.
- If the application that you're trying to open is already running, make sure that it doesn't have any open dialog boxes.
- If it's a linked object, make sure that someone else doesn't already have it open.

- If it's a linked object on a form, make sure that the name of the source file hasn't changed or that the source file hasn't been moved. To see the name of the original file, click the object, and then click **OLE/DDE Links** on the **Edit** menu. The original file name appears in the **Links** box. If the file has been renamed or moved, click the link in the **Links** box, and then click **Change Source** to reconnect the link.

[I want to see the content of my object in my form or report, but I see an icon instead.](#)

1. Open the form in [Design view](#) or the report in Design view.
2. Click the icon.
3. On the **Edit** menu, point to the appropriate **Object** command (for example, point to **Worksheet Object** for a Microsoft Excel object), and then click **Convert**.
4. In the **Convert** dialog box, clear the **Display As Icon** check box.

[My picture becomes distorted when I resize it.](#)

To keep a picture's proportions when you size it, use the **PictureSizeMode** property (for background pictures) or the **SizeMode** property (for bound object frames, unbound object frames, and image controls). If the picture is a bitmap, it's best to use the **Zoom** setting rather than the **Stretch** setting. You should also consider using a [metafile](#) rather than a bitmap since you can size metafiles without causing distortions.

[How can I improve the color quality in my picture on on my form?](#)

If the picture you're displaying was created on a computer that has more colors than the computer that you're displaying the picture on, Microsoft Windows substitutes colors to match the original colors. This process is known as dithering. For example, you may be trying to display a picture that was created with 256 colors on a computer that only displays 16 colors (standard VGA). If your computer can display more colors, you can use a form or report's **PaletteSource** property to specify that Microsoft Access use the color palette that was used to create the picture.

If you're using background pictures with forms that have popup forms, it's a good idea to set the **PaletteSource** property of both forms to the same value.

This is true if the form has a background picture and the popup form doesn't, if the popup form has a background picture and the form doesn't, or if both have background pictures. Windows can use only one palette at a time, so if you set the **PaletteSource** property to the same value, then you ensure that the colors you need to paint each form are available when the focus moves to the popup form or back to the form.



[Why did the size of my database increase significantly more than the size of the object I added?](#)

The amount of information that is used to represent an object is often greater than the size of the object itself. One way to work around this disproportion is to display an icon in the form or report that represents the object instead of the object itself. When you display an icon, the OLE server (the application used to create the object) sends only the information needed to display the icon. When you double-click the icon, the OLE server is launched with the data that the icon represents.

[Show All](#)

Create a label

Use this procedure to create a stand-alone label (a label that isn't attached to another [control](#)). To create a label that's attached to a control, just create the control. Microsoft Access automatically attaches a label to the control when you create it.

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click the **Label** tool  in the [toolbox](#).
3. On the form, report, or data access page, click where you want to place the label, and then type the text for the label.

[Tips](#)

- If you want to display the text in a label on more than one line on a form or report, you can resize the label after you enter all the text, or you can press CTRL+ENTER at the end of each line of text to enter carriage returns. If you insert a carriage return, Microsoft Access automatically wraps subsequent lines as you type. The maximum width of the label is determined by the length of the first line of text.
- If you want to use an ampersand (&) in a label on a form or report, you must type two ampersands. For example, if you want the text "Products & Suppliers" to appear in a label, type **Products && Suppliers**. The extra ampersand is necessary because Access uses a single ampersand in a label or button to define an [access key](#).

[Show All](#)

Create a text box

How you create a text box depends on whether you want it to be [bound](#), [unbound](#), or [calculated](#).

 [Create a bound text box](#)

Before creating a [bound](#) control, make sure that your form or report is bound to a [record source](#), or that your [data access page](#) is connected to a database.

1. Open a form, report, or data access page in [Design view](#).
2. If the [field list](#) isn't visible, click **Field List**  on the [toolbar](#).
3. Click the **Text Box** tool  in the [toolbox](#).
4. Do one of the following:

Select one or more fields in the field list on a form or report.

To select	Do this
One field	Click the field.
A block of fields	Click the first field in the block, hold down the SHIFT key, and then click the last field in the block.
Nonadjacent fields	Hold down the CTRL key and click the name of each field that you want to include.

Select a field, table, or query in the field list on a data access page.

To select	Do this
One field	Click the field.
A table or query	Click the Wizard tool  in the toolbox if it's not already selected , and then click the table or query.

5. Do one of the following:
 - Drag the field or fields from the field list and position them on the form or report.
 - Drag the field, table, or query from the field list and position it on the data access page. If you are dragging a table or query, click the layout

you want in the **Layout Wizard** dialog box.

Note Microsoft Access won't create the control if you try to add it by clicking the form, report, or data access page instead of dragging the selected field from the field list.

Access places one text box on the form, report, or data access page for each field that you select in the field list. Each text box is bound to one field in the underlying data source. Each text box also has an attached label by default.

6. Size the text box so that it's the appropriate size for the data you want to show.
7. Change the label text if necessary.
8. Switch to [Form view](#), [Datasheet view](#), [Page view](#), or [Print Preview](#) to test the control.

Note If the data in a text box on a form contains many lines of text, you might want to set the **ScrollBars** property to **Vertical**.

[Create an unbound text box](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click the **Text Box** tool  in the [toolbox](#).
3. Click anywhere on the form, report, or data access page to create a default-sized text box, or drag to create a text box that's the size you want.

[Create a calculated text box](#)

[In a form or report](#)

1. Open a form or report in [Design view](#).
2. Click the **Text Box** tool  in the [toolbox](#).
3. In the [detail section](#), click where you want to place the text box.
4. Do one of the following:

Place the insertion point in the text box and type an [expression](#) that calculates a total.

Select the text box, click **Properties**  on the toolbar, and type the expression in the **ControlSource** property box. In a [Microsoft Access database \(.mdb\)](#), to use the [Expression Builder](#) to create the expression, click the **Build** button  next to the **ControlSource** property box.

Notes

- If you need more room to type an expression in the **ControlSource** property box, press SHIFT+F2 to open the **Zoom** box.
- If a form or report is based on a [query](#), you may want to put the expression in the query instead of in a [calculated control](#). It may improve performance, and if you are going to calculate totals for groups of records, it's easier to use the name of a [calculated field](#) in an [aggregate function](#).

 [In a data access page](#)

If you want to do complex calculations using spreadsheet functions, use the Microsoft Office Spreadsheet Component. For all other calculations, you can use the [bound span control](#).

 [Using the Spreadsheet Component](#)

1. Open the data access page in [Design view](#).
2. If they aren't already on the page, add the controls that will contain the data you want to use in the calculation.
3. Click the **Office Spreadsheet** tool  in the [toolbox](#), and then click where you want the upper-left corner of the spreadsheet to be on the data access page.
4. In any cell in the spreadsheet, enter a formula that uses the HTML **Host** function to refer to the fields used in the calculation, and the specific operation you want to perform on the field's values expressed as a combination of operators, identifiers, and values.

Example To multiply the value in the Quantity field by the value in the UnitPrice field, type:

=Host().Quantity.value*Host().UnitPrice.value

Example To multiply the value in the UnitPrice field by a percentage rate

stored in cell C1, type:

= $\$C\1 *Host().UnitPrice.value

Notes

- To see the name of a spreadsheet, select the control, and then click **Properties**  on the toolbar. The property sheet title bar displays the spreadsheet name.
- You can use spreadsheet functions, which, in Microsoft Access, are available only in this control.
- Most likely, you'll want the calculated value to display in an area that looks like the other controls on the data access page. To do this you can modify the appearance of the spreadsheet so it looks like a text box.

[Using the bound span control](#)

1. Open the data access page in [Design view](#).
2. If the expression will reference field names, the fields must be in the page's [data definition](#). Either add the fields to the page as [bound span controls](#), or add the fields as bound text boxes.

[Tip](#)

The page will load faster in [Page view](#) and in Microsoft Internet Explorer if you use bound span controls.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

3. Create another control by clicking the **Bound Span** tool  or the **Text Box** tool  in the [toolbox](#), and then click where you want the upper-left corner of the control to be on the page.
4. To display the property sheet, make sure that the control is selected, and

then click **Properties**  on the toolbar.

5. In the **ControlSource** property box, type an alias and an appropriate [expression](#). To reference fields on the page, use the names of the fields in the expression. For example, if you want to compute the extended price for a record in the Order Details table, type:

ExtendedPrice: UnitPrice*Quantity

If you don't specify an alias, you must type an equal sign (=) before the expression. Microsoft Access will automatically supply an alias— for example, Expr1.

[Show All](#)

Attach a label to a control

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click the label that you want to attach to a control.
3. Click **Cut**  on the [toolbar](#).
4. Click the control that you want to attach the label to.
5. Click **Paste**  on the toolbar.

[Show All](#)

Set defaults for labels attached to new controls

You can use this procedure to change the [default control style](#) for the current form or report. This procedure does not apply to data access pages.

1. Open a form or a report in [Design view](#).
2. Click the tool in the toolbox that you want to set defaults for. For example, if you want to specify whether labels are automatically attached to text boxes, click the **Text Box** tool .
3. Click **Properties**  on the **Form Design** or **Report Design** toolbar to display the property sheet for the tool.
4. Microsoft Access changes the title bar of the property sheet to indicate that you're setting default properties for that type of control.
5. Do one or more of the following:

[Specify whether new controls automatically have labels](#)

- In the **AutoLabel** property box, click **Yes** or **No**.

[Specify whether the text in a new label ends with a colon](#)

- In the **AddColon** property box, click **Yes** or **No**.

[Change the default placement of new labels](#)

1. In the **LabelX** property box (horizontal axis), type a positive or negative number that specifies the starting point of the label text relative to the upper-left corner of the control that the label is attached to. A negative number places the label to the left of the control. A positive number places the label to the right of the control.
2. In the **LabelY** property box (vertical axis), type a number. A negative number places the label above the control. A positive number places the label below the control.

For information about how the **LabelX** and **LabelY** properties interact with the **LabelAlign** property, click the **LabelX** or **LabelY** box and press F1.



Tip

To use a unit of measure that is different from the setting specified by the regional settings in Microsoft Windows Control Panel, specify that type of measurement after the number (for example, 1.5 in or 2 cm).

[Show All](#)

Change text in a label or command button

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click the label or button, and then select the label text or button text.



3. Type the new text.

[Show All](#)

Set the margins within a label or text box

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#), and then click **Properties**  on the [toolbar](#) to display the control's property sheet.
3. Adjust the margins of the control by typing the new margin in the appropriate property box.

Forms and reports

Margin	Property
Left	LeftMargin
Top	TopMargin
Right	RightMargin
Bottom	BottomMargin

Data access pages

Margin	Property
Left	PaddingLeft
Top	PaddingTop
Right	PaddingRight
Bottom	PaddingBottom

Note You can enter margins in either inches (in), centimeters (cm), [pixels \(px\)](#), or [points \(pt\)](#). For example, 5px creates a margin that is 5 pixels from the edge of the control.

[Show All](#)

Display text vertically on a form or report

1. Open a form or report in [Design view](#).
2. Select the text box or label, and then click **Properties**  on the [toolbar](#) to display the control's property sheet.
3. In the **Vertical** property box, click **Yes**.

[Show All](#)

Create and customize a scrolling text control

[Create a scrolling text control](#)

1. Open a data access page in [Design view](#).
2. Click the **Scrolling Text** tool  in the [toolbox](#).
3. Do one of the following:

[Create a bound control that displays data](#)

1. Click **Field List**  on the toolbar to display the [field list](#).
2. From the field list, drag the selected field to the data access page. Place the upper-left corner of the icon where you want the upper-left corner of the scrolling text control to be.

[Create an unbound control](#)

1. In the data access page, click where you want the upper-left corner of the scrolling text control to be; Microsoft Access creates a default-sized control. To create a scrolling text control of a particular size, click the page, and then drag until the control is the size you want.
2. Click inside the scrolling text control, and select the default text that is displayed in the control.
3. Type the message that you want to appear.

[Customize a scrolling text control](#)

1. Open a data access page in [Design view](#).
2. Select the scrolling text control, and then click **Properties**  on the toolbar to display the control's property sheet.
3. Do one or more of the following:

[Change the speed](#)

1. In the **TrueSpeed** property box, click **True**.
2. In the **ScrollDelay** property box, type the number of milliseconds to delay between each repetition of the scrolling text.
3. In the **ScrollAmount** property box, type the number of [pixels](#) you want the scrolling text to move in the time that you specified in the **ScrollDelay** property box. The **ScrollAmount** property sets the spacing between each repetition of text.

Note When the **TrueSpeed** property is set to **True**, scrolling text speed is determined by the **ScrollDelay** and **ScrollAmount** properties. For example, if the **ScrollDelay** property is set to **65** and the **ScrollAmount** property is set to **10**, the scrolling text advances 10 pixels every 65 milliseconds. When the **TrueSpeed** property is set to **False**, the shortest delay between repetitions of text is 60 milliseconds.

[Change the motion](#)

- Do one of the following:
 - To make the text scroll continuously in the control like a stock ticker, set the **Behavior** property to **Scroll**.
 - To make the text slide from the starting point to the opposite edge of the control and then remain on the screen, set the **Behavior** property to **Slide**.
 - To make the text go back and forth between the starting point and the opposite edge of the control, and have it remain on the screen at all times, set the **Behavior** property to **Alternate**.

[Change the number of times that scrolling text repeats](#)

- Do one of the following:
 - To make the text scroll continuously, type **-1** in the **Loop** property box.
 - To make the text scroll a specific number of times and then disappear, type an integer in the **Loop** property box. For example, if you set the **Loop** property to **5**, the text will scroll five times.

[Change the direction](#)

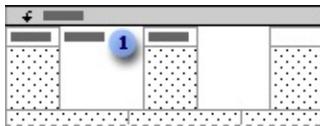
- In the **Direction** property box, click the value that indicates the direction in which you want the text to scroll.

[Show All](#)

Examples of text boxes that grow or shrink when printing forms and reports

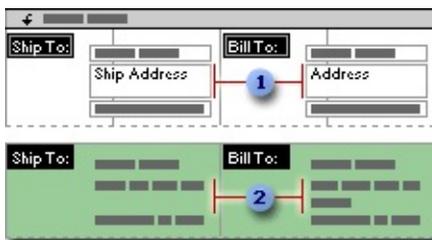
You can make a text box adjust vertically to the size of the data in the field for each record in a form or report. For example, a Memo field can contain values that are very long. You can make the text box large enough to accommodate most of the entries and set **CanGrow** to **Yes**, so that when the form or report is printed, the text box grows to fit longer entries. If a field has no values in some records, you can avoid blank lines by setting **CanShrink** to **Yes**.

When a text box grows or shrinks, the entire section grows or shrinks vertically across its entire width. For that reason, the placement and height of the text box in relation to the [controls](#) beside it are important.



1 To make the text in the text box fill the available space before it grows, make the text box the same height as the tallest neighboring control.

A text box can't shrink to a height that's less than the other controls on the same line. For example, if you have a label beside a text box, the label will always have a value (its caption) even when the text box doesn't have a value. The text box will shrink only to the same height as the label.



1 Text boxes with their **CanShrink** properties set to **Yes** will shrink only as much as other controls on the same line.

2 The Ship To address can't shrink more than the Bill To address.

The Catalog and Invoice reports in the Northwind sample database have examples of text boxes that grow. To view these reports, open the Northwind database in your Office folder's Samples folder, and then open the report you want to see in Design view.

[Show All](#)

Make a text box grow or shrink to fit data when printing a form or report

1. Open the form or report in [Design view](#).
2. Select the text box, and then click **Properties**  on the toolbar.
3. Set the **CanGrow** or **CanShrink** property to **Yes**.

CanGrow expands the text box vertically to fit the data in the field when the form or report is printed.

CanShrink reduces the text box vertically if there is no data in the field when the form or report is printed.

Note When you set the **CanGrow** property to **Yes** for a text box, Microsoft Access automatically sets the **CanGrow** property of the section to **Yes**.

[Show All](#)

Create a check box, toggle button, or option button

How you create a check box, option button, or toggle button depends on whether you want it to be a [bound](#) control or an [unbound](#) control. To create a bound control, you must be working in a form or report that is bound to a [record source](#), or in a [data access page](#) that is connected to a database. In a [Microsoft Access database](#), you can create a control that is bound to a Yes/No field; in a [Microsoft Access project](#), you can create a control that is bound to a **Bit** column.



[Create a bound check box, toggle button, or option button](#)

1. Open a form, report, or data access page in [Design view](#).
2. Click the appropriate tool in the [toolbox](#). If you're not sure which tool to use, place the pointer over the tool until the [ToolTip](#) appears.

Note The toggle button is not available in data access pages.

3. If the [field list](#) isn't visible, click **Field List**  on the toolbar.
4. Click the appropriate field in the field list, and drag it to the form, report, or data access page.

Note The control won't be bound if you try to create it by clicking the form, report, or data access page instead of dragging the selected field from the field list.

5. Change the label text if necessary.
6. Switch to [Form view](#), [Datasheet view](#), [Page view](#), or [Print Preview](#) to test the control.



[Create an unbound check box, toggle button, or option button](#)

1. Open a form, report, or data access page in [Design view](#).
2. Click the appropriate tool in the [toolbox](#). If you're not sure which tool to use, place the pointer over the tool until the [ToolTip](#) appears.

Note The toggle button is not available in data access pages.

3. On the form, report, or data access page, click where you want to place the control. If you're placing the control in an [option group](#) on a form or report, Microsoft Access highlights the option group when you move the pointer over it to show that the control will become part of the option group.

[Show All](#)

Create an option group

You can create an [option group](#) on your own or, if you are working in a form or report, you can use a wizard to create an option group. A wizard speeds up the process of creating an option group because it does all the basic work for you. When you use a wizard, Microsoft Access prompts you for information, and then creates an option group based on your answers.

In a [data access page](#), you use option buttons to represent the different values that you can save in the underlying field of the option group. In a form or report, you can use check boxes, option buttons, or toggle buttons to represent the different values that you can save in the underlying field.

[Create an option group by using a wizard](#)

1. Open a form or report in [Design view](#).
2. Click the **Control Wizards** tool  in the [toolbox](#) if it's not already [selected](#).
3. In the toolbox, click the **Option Group** tool .
4. On the form or report, click where you want to place the upper-left corner of the option group.
5. Follow the directions in the wizard dialog boxes. In the last dialog box, click **Finish** to display the option group in Design view.

[Create an option group on my own](#)

1. Open a form, report, or data access page in [Design view](#).
2. Click the **Control Wizards** tool  in the [toolbox](#) if it's [selected](#). This turns off the wizard.
3. In the toolbox, click the **Option Group** tool .
4. Do one of the following:

To create an [unbound](#) option group, click where you want to place the group frame.

To create a [bound](#) option group, click **Field List**  on the [toolbar](#) to

display the [field list](#), and then drag the appropriate field or column from the field list to the form, report, or data access page.

Note The control won't be bound if you try to create it by clicking the form, report, or data access page instead of dragging the selected field from the field list.

5. In the form or report toolbox, click the **Check Box** , **Option Button** , or the **Toggle Button** tool . In the data access page toolbox, click the **Option Button**  tool.
6. Click inside the group frame where you want the upper-left corner of the check box, option button, or toggle button to appear.
7. Do one of the following:

On a form or report, if necessary, click the control, click **Properties**  on the toolbar, and then change the **OptionValue** property to the value that you want the option group to have when you click that control. When you add the first control to an option group on a form or report, Microsoft Access sets its **OptionValue** property to **1**.

On a data access page, click the option button, click **Properties**  on the toolbar, and then change the **Value** property from **on** to the value that you want the option group to have when you click that button.

If the option group is bound, the value of the control's **OptionValue** or **Value** property is the value that Microsoft Access stores in the underlying table when you click that control.



1 OptionValue or **Value** property set to **1** - clicking this button sets the value of the Ship Via option group to 1 and, if the option group is bound, stores that value in the underlying table.

2 OptionValue or **Value** property set to **2** - clicking this button sets the value of the Ship Via option group to 2 and, if the option group is bound, stores that value in the underlying table.

3 **OptionValue** or **Value** property set to 3 - clicking this button sets the value of the Ship Via option group to 3 and, if the option group is bound, stores that value in the underlying table.

8. Repeat steps 5 through 7 for each control you add to the option group. On a form or report, Access sets the **OptionValue** property of the second control to 2, the third to 3, and so on.



To facilitate data entry, you can make the most commonly chosen option the default by setting the **DefaultValue** property of the option group.

[Show All](#)

Move a control to an option group

If you create a check box, option button, or toggle button outside an [option group](#), and you want to add that control to an option group, you must cut and paste the control into the option group. If you drag an existing control into a group frame, the control doesn't become part of the option group.

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the control you want to move to the option group.
3. Click **Cut**  on the [toolbar](#).
4. Select the frame of the option group, and then click **Paste**  on the toolbar.
5. Select the control, and click **Properties**  on the toolbar.
6. Do one of the following:

If the control is on a form or report, change the **OptionValue** property from **-1** to the value that you want the option group to have when you click the control.

If the control is on a data access page, change the **Value** property from **on** to the value that you want the option group to have when you click the control.

[Show All](#)

About creating a list box, combo box, or drop-down list box that looks up values

To create a list box, [drop-down list box](#), or combo box that looks up values, you need to decide two things:

- Where will the rows for the list come from?
- When a user selects an item in the list box, drop-down list box, or combo box, what will Microsoft Access do with that value?

The rows for a list box, drop-down list box, or combo box can come from a fixed list of values that you enter when you create the list box, drop-down list box, or combo box (use when the list won't change very often), or they can come from a [record source](#) (use when you'll frequently update the list). For example, you could use a fixed list for a list box that contains entries for Mr., Mrs., or Ms. If, on the other hand, you have a Products form and you want to display a list of product suppliers in a list box, but the list will change frequently, base the list box on a record source. The list box looks up values in the Suppliers table and displays an up-to-date list of the suppliers that you can choose from.



- 1 Enter a product.
- 2 Pick a supplier from the list.

When a user selects a value in a list box, drop-down list box, or combo box, Microsoft Access can do one of two things: Access can store the selected value in a table (not the same table that the list gets its rows from), or Access can pass the value to another [control](#). For example, for the Supplier list box in the preceding illustration, if a user selects "Pavlova, Ltd." , Access looks up the

[primary key](#) value (SupplierID) for Pavlova, Ltd. in the Suppliers table and sets the SupplierID field (the [foreign key](#)) for the current record in the Products table to the same value. This is the value that is stored. Because Access is storing a value based on a selection in the list box, the list box is [bound](#). (Note that the SupplierName value that comes from the Suppliers table is displayed in the list box but not stored.)

If, on the other hand, you want Access to use the value selected in the list box, drop-down list box, or combo box to determine the contents of another control or controls, you would create an [unbound](#) list box, drop-down list box, or combo box. For example, you might want to create a list box or drop-down list box that you can use to find a related record on a [data access page](#). Or you might want to create a list box or combo box that you use to filter the records in another list box or combo box on a form.

[Should I create a list box, a combo box, or a drop-down list box?](#)

To decide whether to create a list box, combo box, or drop-down list box, think about where you're going to use the [control](#), how you want the control to appear on your form or data access page, and how you expect people to use it. Each control offers advantages:

Advantages of a list box You can use list boxes on forms and data access pages. The list appears at all times and the value of the control is limited to the set of alternatives in the list. To move quickly to the first value that starts with a particular letter, you can type that letter. When you're using a form to enter or edit data, you can't add a value that's not in the list.

Advantages of a combo box You can use combo boxes on forms. Since the list isn't displayed until you open it, the control uses less room on the form. You can type the first few characters of a value in the combo box to quickly find that value. You can also control whether any value can be entered in the list, or only text that matches one of the values in the list can be entered.

Advantages of a drop-down list box You can use drop-down list boxes on data access pages. Because the list isn't displayed until you open it, the control uses less room on the data access page. The value of the control is limited to the set of alternatives in the list. To move quickly to the first value that starts with a particular letter, you can type that letter.

[Should I use a table, a query, an SQL statement, or a function to provide the values?](#)

The row source of a list box, combo box, or drop-down list box provides the values that are displayed. In most cases, you can use a table as the row source when you create the list box, drop-down list box, or combo box with a wizard. Use a [query](#) in the following cases:

- If you want to include calculated fields in the list box, drop-down list box, or combo box. For example, you could concatenate a first name and a last name by typing the following expression: **Name: [FirstName] & " "& [LastName]**. You could define this calculated field in a query, and then include this field in the list box, drop-down list box, or combo box.
- If you want to include fields from more than one table in the list box, combo box, or drop-down list box.

When you select fields from a table or query by using a wizard in a form, Microsoft Access creates an [SQL statement](#) based on the fields that you select and sets the **RowSource** property of the list box or combo box to that SQL statement. When you select fields from a table or query by using a wizard in a data access page, Access sets the **ListRowSource** property of the list box or drop-down list box to the table or query you selected.

You can also set the row source of a list box, combo box, or drop-down list box yourself. For example, you can set the row source to a table or a saved query, or you can set the row source to an SQL statement so that the query doesn't show up in the [Database window](#). However, list boxes and combo boxes based on SQL statements are slower than list boxes and combo boxes based on saved queries.

In addition, you can use a [function](#) as a type of row source when the values in a list are constantly changing, or if the values aren't stored in a database. For example, you could write a function that fills a list box with a list of dates, each one a specific number of days after the current date.

[Ways to create a list box, combo box, or drop-down list box](#)

Wizards Using a wizard is an easy way to create a list box, combo box, or drop-down list box. However, wizards are not available in a stand-alone data access page, or when you point a data access page to another database than the

one currently open. In this case, you will have to create a list box or drop-down list box on your own.

Lookup fields You can create a list box, combo box, or drop-down list box that is [bound](#) to a [Lookup field](#) in a [Microsoft Access database](#) or in a [Microsoft Access project](#) that is connected to a database in Microsoft SQL Server 2000 or the Microsoft Access SQL Server Desktop Engine. When you create a Lookup field, you only need to create the list box or combo box once. Then you can use the same lookup list on any form. If you don't plan to use the same lookup list in more than one form, you can use the List Box Wizard or Combo Box Wizard in form [Design view](#). You can't use a Lookup field to create a list box or drop-down list box in a data access page.

On your own You might want to create a list box, combo box, or drop-down list box without using a wizard or lookup field if you want to set the properties of the control yourself. For example, you might not want to use a wizard or lookup field if you want to specify that the type of row source is a function.

[Show All](#)

Create a list box, combo box, or drop-down list box

A list box, combo box, or [drop-down list box](#) can be a [bound](#) or [unbound](#) control, and it can look up values in a fixed list or in a table or query. To create a bound list box, combo box, or drop-down list box that looks up values in a table or query, make sure the form or data access page is based on a [record source](#) that includes a [foreign key](#) field or [Lookup field](#).

 [Create a list box, combo box, or drop-down list box by using a wizard](#)

1. Open a form or data access page in [Design view](#).
2. In the [toolbox](#), make sure that the **Control Wizards** tool  is [selected](#).
3. Click the **List Box** tool , the **Combo Box** tool , or the **Dropdown List** tool  in the toolbox.
4. On the form or data access page, click where you want to place the list box, combo box, or [drop-down list box](#). In the wizard dialog boxes, follow the instructions and choose the options you want.

 [How?](#)

1. When the wizard asks how you want to get the values for the control, do one of the following:
 - If you want to display a fixed list of values that will seldom change, click **I will type in the values that I want**.
 - If you want to display the current data in a record source, click **I want the list box to look up the values in a table or query**.
2. Follow the instructions for specifying how the values will appear.
3. If you are creating a list box or combo box in a form, when the wizard asks what you want Microsoft Access to do when you select a value, do one of the following:
 - If you are creating an unbound control, click **Remember the value for later use**.
 - If you are creating a bound control, click **Store that value in this**

field and select the field you want.

4. Click **Finish**.
5. In a data access page, click **Properties**  on the **Page Design** toolbar to open the property sheet for the control. In the **ControlSource** property box, select the field you want to bind the list box or the drop-down list box to.

 [Create a list box, combo box, or drop-down list box that is bound to a Lookup field in a table](#)

You can use a [Lookup field](#) in a table to create a [bound](#) list box, combo box, or [drop-down list box](#) in a [Microsoft Access database](#) or an [Access project](#) that is connected to a database in Microsoft SQL Server 2000 or the Microsoft SQL Server 2000 Desktop Engine.

1. In [Design view](#) of a table, create the Lookup field.
2. Do one of the following:

 [Create a new form with a list box or combo box that is bound to the Lookup field](#)

- Create a new form that is based on a [record source](#) that includes the Lookup field. Access automatically creates the lookup list box or combo box.

 [Create a new data access page with a list box or drop-down list box that is bound to the Lookup field](#)

1. Create a new data access page in Design view. The page must be connected to the database containing the Lookup field.
2. If the field list isn't displayed, click **Field List**  on the **Page Design** toolbar.
3. Drag the table or query that contains the Lookup field from the field list to the data access page. When the Layout Wizard appears, select **Columnar** or **Tabular**. Access automatically creates the lookup list box or drop-down list box.

 [Add a list box, combo box, or drop-down list box to a form or data](#)

[access page](#)

1. In Design view, open a form that is based on a [record source](#) that includes the Lookup field, or open a data access page that is connected to the database that includes the Lookup field.
2. If the field list isn't displayed, click **Field List**  on the **Form Design** or **Page Design** toolbar.
3. Drag the Lookup field from the field list to the form or data access page. Access automatically creates the lookup list box, combo box, or drop-down list box.

[Create a list box, combo box, or drop-down list box from scratch](#)

When you create a list box, combo box, or drop-down list box from scratch, you set many of the properties of the control yourself. If you want more information about a particular property, click the appropriate property box and press F1.

1. Open a form or data access page in [Design view](#).
2. In the [toolbox](#), make sure that the **Control Wizards** tool  is not [selected](#).
3. Click the **List Box** tool , the **Combo Box** tool , or the **Dropdown List** tool  in the toolbox.
4. Click the the form, report, or data access page to create a default-sized control, or click and drag until the control is the size you want.
5. With the control still selected, click **Properties**  on the **Form Design** or **Page Design** toolbar to open the property sheet for the control.
6. Do one of the following:

[Set properties for a list box or combo box in a form](#)

1. In the **RowSourceType** property box, do one of the following:
 - In an Access database, to show values from a table or query, or results of an SQL statement, select **Table/Query**.
 - In an Access project, to show values from a table or query, or results of an SQL statement, select **Table/View/StoredProc**.
 - To show a fixed list of values, select **Value List**.
 - To show a list of fields in a table or query, select **Field List**.
2. In the **RowSource** box, do one of the following:
 - Select a table or query containing the values or field names that

- you want to appear in the list box.
- Type a list of fixed values and separate each item with a semicolon (;).
 - Type an SQL statement or click the **Build**  button to open the Query Builder.
3. If you want more than one column to appear in the control, click the **ColumnCount** property box and type the number of columns you want.

 [Set properties for a list box or drop-down list box in a data access page](#)

1. Set the **ListRowSource** property box, select the recordset you want.
2. In the **ListBoundField** and **ListDisplayField** property boxes, select the fields you want.
7. If you want Access to store the value you select, click the **ControlSource** property box and select the field to which you want to bind the list box, combo box, or drop-down list.

 [In a form, create a list box or combo box that gets its rows from a Visual Basic function](#)

You may want to use this procedure when the values in a list are constantly changing, or if the values aren't stored in a database. For example, you could write a function that fills a list box with a list of dates, each one a specific number of days after the current date.

1. Create a [Visual Basic function](#).
2. In the [toolbox](#), make sure that the **Control Wizards** tool  is not [selected](#).
3. Click the **List Box** tool  or the **Combo Box** tool  in the toolbox.
4. With the control still selected, click **Properties**  on the **Form Design** toolbar to open the property sheet for the control.
5. In the **RowSourceType** property box, enter the name of the function. Don't put an equal sign before the function name.
6. Leave the **RowSource** property box blank.

 [Customize a list box, combo box, or drop-down list box](#)

1. In [Design view](#) of the form or data access page, make sure that the [list box](#), [combo box](#), or [drop-down list box](#) is selected, and then click **Properties**  on the toolbar to open the property sheet for the [control](#).
2. Do one of the following:

Change the sort order in a list box, combo box, or drop-down list box

If you used a wizard to create the list box, drop-down list box, or combo box, Microsoft Access automatically sorts the rows that make up the list by the first visible column. If you want to specify a different sort order, or if you have set the **RowSource** property of the control to a saved query, you can use one of the following two procedures.

Note Wizards are not available in a stand-alone data access page, or in a data access page that is connected to a database other than the one currently open. In this case, you will have to create a list box or drop-down list box on your own.

- To change the sort order of a control on a form, click the **Data** tab and next to the **RowSource** property box, click the **Build** button  to open the Query Builder or SQL Statement Builder and specify the sort order you want.
- To change the sort order of a control on a data access page, create a query with the sort order you want, and then in the data access page's **ListRowSource** property box, select the query from the list.

Bind a column from a list box, combo box, or drop-down list box

- **In a form** In the **BoundColumn** property box of the list box or combo box, specify a number that corresponds to the placement of the column in the list box or combo box. For example, type **1** to bind the first column in the list box or combo box to the underlying field specified in the **ControlSource** property. Include hidden columns when you count columns.

If you set the **BoundColumn** property to **0**, Microsoft Access saves

the [list index](#) instead of a value from one of the columns. This is useful if you want to store a sequence of numbers instead of the list value.

- **In a data access page** In the **ListBoundField** property box of the list box or drop-down list box, enter the name of the field that corresponds to the placement of the column in the list box or drop-down list box.

Use an SQL statement as the row source for a list box or combo box on a form

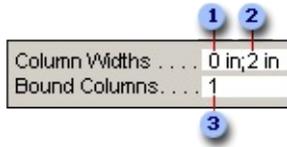
You might want to use an [SQL statement](#) instead of a saved query for the [row source](#) of a list box or combo box on a form so that the query doesn't show up in the [Database window](#). However, list boxes and combo boxes based on SQL statements are slower than list boxes and combo boxes based on saved queries.

1. Click the **Build** button  next to the **RowSource** property to open the Query Builder.
2. In the Query Builder, click **SQL View** on the **View** menu.
3. In the SQL window, select the entire SQL statement, and then press CTRL+C to copy it. Close the Query Builder.
4. Click in the **RowSource** property box, and press CTRL+V to paste the SQL statement into the property box.
5. Delete the original query in the Database window.

Hide a column in a list box or combo box on a form

- In the **ColumnWidths** property box, enter **0** for the column or columns that you want to hide.

For example, suppose you have a bound two-column combo box that has a SupplierID column and a SupplierName column. The SupplierID column is the first column in the list. To hide the SupplierID column, set the **ColumnWidths** property as shown in the following illustration.



- 1 Set the SupplierID column (first column) to 0 to hide it.
- 2 Specify a width for the SupplierName column.
- 3 The SupplierID column is the bound column, even though it's hidden.

Note In a combo box, the first visible column is displayed in the text box portion of the combo box when the list isn't displayed. For example, the SupplierName column in the previous example would be displayed because the SupplierID column is hidden. If the SupplierID column weren't hidden, it would be the only column that was displayed.

Add column headings to a combo box on a form

- o In the **ColumnHeads** property box, click **Yes** to display column headings. Headings in combo boxes appear only when the list is open.

If the combo box or list box is based on a [record source](#), Microsoft Access uses the field names from the record source as the column headings. If the combo box or list box is based on a fixed value list, Microsoft Access uses the first x items of data from the value list (**RowSource** property) as the column headings, where x = the number set in the **ColumnCount** property.

Turn off the fill-in-as-you-type feature for a combo box on a form

- o In the **AutoExpand** property box, click **No**.

When the **AutoExpand** property is set to **No**, you must select a value from the list or type the entire value.

Set the width of the list box portion of a combo box on a form

- In the **ListWidth** property box, enter the desired width using the current unit of measurement (set in Windows Control Panel). To use a unit of measurement other than the default, include a measurement indicator. For example, enter **2 cm**. Make sure to leave enough space for a scroll bar.

The list box portion of the combo box can be wider than the text box portion, but it can't be narrower. The default setting (**Auto**) makes the list box the same width as the text box portion of the combo box.

Set the maximum number of rows to display in a combo box on a form

- In the **ListRows** property box, enter a number.

If the actual number of rows exceeds the number specified in the **ListRows** property, a vertical scroll bar is displayed in the combo box.

Limit combo box entries to items in the list portion of a combo box on a form

- In the **LimitToList** property box, click **Yes**.

Notes

If the first column displayed in a combo box is not the [bound](#) column, Microsoft Access limits the entries to the list even if the **LimitToList** property is set to **No**.

If the **LimitToList** property is set to **No**, when you enter an entry that isn't in the list, if the combo box is bound, the entry is stored in the underlying field, but it isn't added to the list. To add new entries to the list, use the **OnNotInList** property and the **NotInList** event.

[Show All](#)

Specify which row is the default value in a list box or combo box

In a form, you can specify which row is automatically selected in a list box or combo box. This procedure does not apply to data access pages.

1. Open a form in [Design view](#).
2. Make sure that the combo box is selected and then click **Properties**  on the **Form Design** toolbar.
3. In the **DefaultValue** property box, type **[comboboxname].ItemData(*n*)** where *n* is the row that you want to use as the default. The **ItemData** property is zero-based, so type **ItemData(0)** if you want to make the first row in the list the default. For example, to specify the first row of the SupplierCombo combo box as the default for the combo box, type **[SupplierCombo].ItemData(0)** in the **DefaultValue** property box.

Note If you want to specify a default value for a lookup combo box in [Microsoft Access database \(.mdb\)](#) table [Design view](#), enter a default value in the **DefaultValue** property for the [foreign key](#). For example, if you have a SupplierID foreign key in a Products table, click the SupplierID field, and then enter a valid supplier ID number in the **DefaultValue** property box in the lower portion of the window. The next time you add this Supplier ID field to a form, Access automatically sets the default value for the field.

[Show All](#)

Find a record by selecting a value from a list

On a form, you can create a [list box](#) or a [combo box](#) that finds a record when you select a value from the list.

1. Open the form in [Design view](#).
2. Click **Control Wizards**  in the [toolbox](#) if it's not already pressed in.
3. In the toolbox, click **List Box**  or click **Combo Box** .
4. On the form, click where you want to place the list box or combo box.
5. In the first wizard dialog box, click the option to find a record based on the value you selected in the list box or combo box.
6. Follow the directions in the wizard dialog boxes. In the last dialog box, click **Finish** to display the list box or combo box in Design view.

If you want to see the [event procedure](#) that makes the list box or combo box work, open the box's property sheet, and then click the **Build** button  next to the **AfterUpdate** property box.

[Show All](#)

Properties of list boxes, combo boxes, drop-down list boxes, and Lookup fields

 [Properties you can use to create or modify list boxes and combo boxes on a form](#)

When you use a wizard to create a list box or combo box, Microsoft Access sets certain properties for the [control](#). You can work with these properties directly if you don't want to use a wizard to create the control, or you can use a wizard to create the control and then modify these properties to make the control work just the way you want. To find out more about any of the following properties, click the property in the property sheet and press F1.

In an [Access database](#), or in an [Access project](#) connected to Microsoft SQL Server 2000 or Microsoft SQL Server 2000 Desktop Engine, you can also use these properties to customize a [Lookup field](#).

Property	Description
RowSourceType	Works together with the RowSource property. You use this property to specify the type of row source (Table/Query , Table/View/StoredProc , Value List , Field List , or Visual Basic function), and then you specify the actual row source in the RowSource property.
RowSource	Specifies the name of the table, query, or SQL statement if RowSourceType is set to Table/Query . Specifies the entries in the list, separated by semicolons, if RowSourceType is set to Value List . Specifies the name of the table or query if RowSourceType is set to Field List . Is blank if RowSourceType is set to a function name.
	In a bound multiple-column list box or combo box, specifies which column is bound to the underlying field specified in the ControlSource property. Data in that column will be stored in

BoundColumn	the field when you select that item in the list. This data may be different from the data that appears in the list if you hide the column.
ColumnCount	Specifies the number of columns in a list box or combo box. You can include a column but not display it in a list by setting the ColumnWidths property to 0 . You might want to do this if you have a list that looks up values from a table or query. You could include an ID field but hide it in the list. In a combo box, the first visible column in the list appears in the text box portion of the combo box.
ColumnWidths	Specifies the width of each column, separated by semicolons. Enter 0 to hide a column. Enter a semicolon without a measurement to use the default (about 1 in. or 2.5 cm, depending on the unit of measurement set in Windows Control Panel). In a combo box, the first visible column appears in the text box portion of the control. The data type of any value you type in the combo box must be the same as or compatible with the data type of the first visible column.
ColumnHeads	Determines whether the list box or combo box has column headings. If the combo box or list box is based on a record source , Access uses the field or column names from the record source as the column headings. If the combo box or list box is based on a fixed value list, Access uses the first x items of data from the value list (RowSource property) as the column headings, where x = the number set in the ColumnCount property. Headings appear in combo boxes only when the list is dropped down.
ListWidth	Specifies the width of the list box portion of a combo box.
ListRows	Specifies the maximum number of rows to display in the list box portion of a combo box.
LimitToList	Determines whether a combo box accepts any entered text or only text that matches one of the values in the list. If you want to allow a new value that a user enters to be added to the list, set the property to No and attach an event procedure to the OnNotInList property.
	Runs the attached event procedure when the NotInList event

OnNotInList	occurs. You can use this property to automatically add a new value to a list. This property isn't available on the Lookup tab in table Design view.
AutoExpand	Specifies whether Access automatically fills in a value that matches the characters that you type in a combo box. This property isn't available on the Lookup tab in table Design view— Access automatically fills in a value for a Lookup field. However, you can change the AutoExpand property of a combo box based on a Lookup field.
DisplayControl	Specifies the type of control (combo box or list box) that Access creates by default when you add a Lookup field to a form. This property is available only on the Lookup tab in table Design view.

 [Properties you can use to create or modify a list box or a drop-down list box in a data access page](#)

When you use a wizard to create a list box or a [drop-down list box](#) in a [data access page](#), Access sets certain properties for the [control](#). You can work with these properties directly if you don't want to use a wizard to create the control, or you can use a wizard to create the control and then modify these properties to make the control work just the way you want. To find out more about any of the following properties, click the property in the property sheet and press F1.

Property	Description
ListRowSource	The name of a record set in the data source that supplies the values and text for the list box or drop-down list box.
ListBoundField	In a bound list box or drop-down list box, specifies which field is bound to the underlying data source specified in the ListRowSource property. Data in that column will be stored in the field when you select that item in the list.
ListDisplayField	In a bound list box or drop-down list box, specifies the name of a field in the ListRowSource that provides the display values that correspond to the ListBoundField values.

[Show All](#)

Customize a list box, combo box, or drop-down list box

1. In [Design view](#) of the form or data access page, make sure that the [list box](#), [combo box](#), or [drop-down list box](#) is selected, and then click **Properties**  on the toolbar to open the property sheet for the [control](#).
2. Do one of the following:

Change the sort order in a list box, combo box, or drop-down list box

If you used a wizard to create the list box, drop-down list box, or combo box, Microsoft Access automatically sorts the rows that make up the list by the first visible column. If you want to specify a different sort order, or if you have set the **RowSource** property of the control to a saved query, you can use one of the following two procedures.

Note Wizards are not available in a stand-alone data access page, or in a data access page that is connected to a database other than the one currently open. In this case, you will have to create a list box or drop-down list box on your own.

- To change the sort order of a control on a form, click the **Data** tab and next to the **RowSource** property box, click the **Build** button  to open the Query Builder or SQL Statement Builder and specify the sort order you want.
- To change the sort order of a control on a data access page, create a query with the sort order you want, and then in the data access page's **ListRowSource** property box, select the query from the list.

Bind a column from a list box, combo box, or drop-down list box

- **In a form** In the **BoundColumn** property box of the list box or combo box, specify a number that corresponds to the placement of the

column in the list box or combo box. For example, type **1** to bind the first column in the list box or combo box to the underlying field specified in the **ControlSource** property. Include hidden columns when you count columns.

If you set the **BoundColumn** property to **0**, Microsoft Access saves the [list index](#) instead of a value from one of the columns. This is useful if you want to store a sequence of numbers instead of the list value.

- **In a data access page** In the **ListBoundField** property box of the list box or drop-down list box, enter the name of the field that corresponds to the placement of the column in the list box or drop-down list box.

Use an SQL statement as the row source for a list box or combo box on a form

You might want to use an [SQL statement](#) instead of a saved query for the row source of a list box or combo box on a form so that the query doesn't show up in the [Database window](#). However, list boxes and combo boxes based on SQL statements are slower than list boxes and combo boxes based on saved queries.

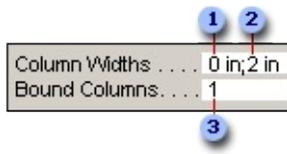
1. Click the **Build** button  next to the **RowSource** property to open the Query Builder.
2. In the Query Builder, click **SQL View** on the **View** menu.
3. In the SQL window, select the entire SQL statement, and then press CTRL+C to copy it. Close the Query Builder.
4. Click in the **RowSource** property box, and press CTRL+V to paste the SQL statement into the property box.
5. Delete the original query in the Database window.

Hide a column in a list box or combo box on a form

- In the **ColumnWidths** property box, enter **0** for the column or columns that you want to hide.

For example, suppose you have a bound two-column combo box that has a SupplierID column and a SupplierName column. The SupplierID

column is the first column in the list. To hide the SupplierID column, set the **ColumnWidths** property as shown in the following illustration.



- 1 Set the SupplierID column (first column) to 0 to hide it.
- 2 Specify a width for the SupplierName column.
- 3 The SupplierID column is the bound column, even though it's hidden.

Note In a combo box, the first visible column is displayed in the text box portion of the combo box when the list isn't displayed. For example, the SupplierName column in the previous example would be displayed because the SupplierID column is hidden. If the SupplierID column weren't hidden, it would be the only column that was displayed.

Add column headings to a combo box on a form

- o In the **ColumnHeads** property box, click **Yes** to display column headings. Headings in combo boxes appear only when the list is open.

If the combo box or list box is based on a [record source](#), Microsoft Access uses the field names from the record source as the column headings. If the combo box or list box is based on a fixed value list, Microsoft Access uses the first x items of data from the value list (**RowSource** property) as the column headings, where x = the number set in the **ColumnCount** property.

Turn off the fill-in-as-you-type feature for a combo box on a form

- o In the **AutoExpand** property box, click **No**.

When the **AutoExpand** property is set to **No**, you must select a value from the list or type the entire value.

Set the width of the list box portion of a combo box on a form

- In the **ListWidth** property box, enter the desired width using the current unit of measurement (set in Windows Control Panel). To use a unit of measurement other than the default, include a measurement indicator. For example, enter **2 cm**. Make sure to leave enough space for a scroll bar.

The list box portion of the combo box can be wider than the text box portion, but it can't be narrower. The default setting (**Auto**) makes the list box the same width as the text box portion of the combo box.

Set the maximum number of rows to display in a combo box on a form

- In the **ListRows** property box, enter a number.

If the actual number of rows exceeds the number specified in the **ListRows** property, a vertical scroll bar is displayed in the combo box.

Limit combo box entries to items in the list portion of a combo box on a form

- In the **LimitToList** property box, click **Yes**.

Notes

If the first column displayed in a combo box is not the [bound](#) column, Microsoft Access limits the entries to the list even if the **LimitToList** property is set to **No**.

If the **LimitToList** property is set to **No**, when you enter an entry that isn't in the list, if the combo box is bound, the entry is stored in the underlying field, but it isn't added to the list. To add new entries to the list, use the **OnNotInList** property and the **NotInList** event.

[Show All](#)

Work with command buttons

Once you have created your command button, you can customize it to better fit your needs.

[Move the command button](#)

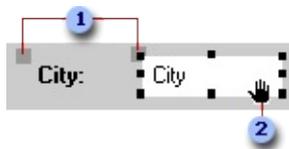
1. Open a form, report, or [data access page](#) in [Design view](#).
2. Do one of the following:

[Move a control and its label together](#)

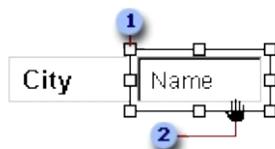
1. Click the [control](#) or its [label](#).

When you click either part of a [compound control](#) in forms and reports, Microsoft Access displays the move handles for both controls, as well as the sizing handles for the control that you clicked. On data access pages, when the control is selected, the label is not automatically selected. If you move the control, the label goes with it. But if you move the label, the control does not move.

2. Move the pointer over the border (not the move handle) of either the control or its label. After the pointer changes to an open hand, you can drag the control or controls to a new position.



- 1 Move handles of a control in a form or report.
- 2 When the pointer changes to an open hand, you can drag the control.



- 1 Move handle of a control in a data access page
- 2 When the pointer changes to an open hand, you can drag the control.

[Move a control and its label separately](#)

In a data access page, you can't move a control separately from its label, but you can select the label and move it separately from the control.

1. Click the [control](#) or its [label](#).
2. Do one of the following:
 - In a form or report, place the pointer over the move handle in the upper-left corner of the control or label, and the pointer changes to an upward-pointing hand.



- 1 When the pointer changes to an upward-pointing hand, you can drag the control.
 - In a data access page, place the pointer on any edge of the label, and the pointer changes to an open hand.
3. Drag the control or label to its new position.

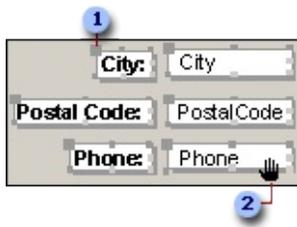
Notes

- The parts of a [compound control](#) remain related, even if you move the individual parts separately.
- To move an attached label to another [section](#) without moving the control, you must use the **Cut** and **Paste** commands. In a form or report, if you move a label to another section, the label will no longer be related to the control.

[Move two or more controls at the same time](#)

1. Select the controls by holding down the SHIFT key as you click each control.
2. Pass the pointer over the border (not the move handle) of any selected

control until it turns into an open hand, and then drag the controls to a new position.



1 Move handle

2 When the pointer changes to an open hand, you can drag the controls.



[Move a control in front of or behind other controls](#)

- Click the [control](#) that you want to move, and then on the **Format** menu, click **Send to Back** or **Bring to Front**.

Note A list box or [subform](#) is always in front of other controls; you can't format a list box or subform so that it is behind another control.



[Tips](#)

- When you move a control in a form or report, you can maintain its horizontal or vertical alignment with other controls by holding down the SHIFT key while selecting and moving the control. You can move the control horizontally or vertically only, depending on which direction you move it first.
- If you need to make small adjustments to the placement of a control, you might find it easier to press the appropriate arrow keys. If **Snap to Grid** is on when you use the arrow keys to move a control, Access aligns the control to the grid.
- If you want to temporarily override the current **Snap to Grid** setting, hold down the CTRL key while you're placing, moving, or sizing a control. For example, if **Snap to Grid** is turned on, you can hold down the CTRL key and move a control anywhere on the form, report or data access page without aligning it to the grid. Releasing the CTRL key restores the current **Snap to Grid** setting.

[Size the command button](#)

[Resize a control](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#) or controls that you want to resize. To select more than one control, hold down the SHIFT key and click each control, or drag a rectangle over the controls.
3. Drag the sizing handles until the control is the size you want.



- 1 Move handle
- 2 Sizing handles

If you select more than one control, all the controls change size as you drag the sizing handle of one control.

Note In data access pages, you can't resize a [drop-down list box](#) vertically. The vertical size of the drop-down list box is based on the font setting of the data in the control.

[Tip](#)

If you want to make small adjustments to the size of a control in a form or report, you might find it easier to press SHIFT plus the appropriate arrow keys.

[Size controls uniformly relative to each other on a form or report](#)

1. Open a form or report in [Design view](#).
2. Select the [controls](#) that you want to size by holding down the SHIFT key and clicking each control, or by dragging a rectangle over the controls.
3. On the **Format** menu, point to **Size**, and then click one of the following commands:
 - **To Tallest** Makes the controls as tall as the tallest selected control.
 - **To Shortest** Makes the controls as short as the shortest selected

control.

- **To Widest** Makes the controls as wide as the widest selected control.
- **To Narrowest** Makes the controls as narrow as the narrowest selected control.

[Size controls uniformly relative to each other on a data access page](#)

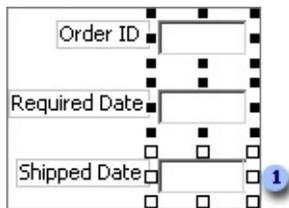
How you size controls relative to each other depends on whether you have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2), or 5.5 or later installed.

- Open a data access page in [Design view](#).

[If you have Internet Explorer 5.5 or later installed](#)

1. Select the [controls](#) that you want to resize by holding down the SHIFT key and clicking each control.

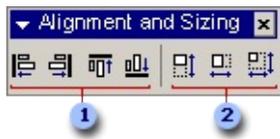
Make sure that the control that you want to align the other controls to is the last control you select and that its sizing handles are white.



1. The sizing handles of the primary selection are white.
2. On the **Format** menu, point to **Size**, and then click one of the following commands:
 - **Height** Makes the controls as tall as the primary selection.
 - **Width** Makes the controls as wide as the primary selection.
 - **Both** Makes the controls as tall and as wide as the primary selection.

[If you have Microsoft Internet Explorer 5.01 with Service Pack 2 \(SP2\) installed](#)

1. On the **Format** menu, click **Alignment and Sizing** to display the **Alignment and Sizing** toolbar.



- 1 **Align Left, Align Right, Align Top, and Align Bottom** buttons
 - 2 **Size Height, Size Width, and Size Height and Width** buttons
2. Make sure the [control](#) that is the size you want is selected.
3. Do one of the following:
 - To resize just one control, click the button that you want on the **Alignment and Sizing** toolbar, and then click the control that you want to resize.
 - To resize several controls, lock the button that you want on the **Alignment and Sizing** toolbar by double-clicking it, and then click the controls that you want to resize.

To release the lock, click the same or a different button on the **Alignment and Sizing** toolbar, or press ESC.

[Resize a control to fit its contents](#)

You can resize a [label](#), [command button](#), [image control](#), or [unbound object frame](#) to fit its contents. Also, if you increase the font size for a control, you can resize the control to accommodate the larger font.

1. Open a form, report, or data access page in [Design view](#).
2. Select the [control](#) or controls that you want to resize.
3. On the **Format** menu, point to **Size**, and then click **To Fit**.

[Resize a control by using the grid](#)

1. Open a form, report, or data access page in [Design view](#).
2. If the grid isn't already displayed, on the **View** menu, click **Grid**.
3. Select the [controls](#) that you want to resize.
4. On the **Format** menu, point to **Size**, and then click **To Grid**.

 [Add a picture to the command button](#)

 [Add a picture to a command button or a toggle button on a form](#)

1. Open a form in [Design view](#).
2. Click the command button or toggle button, and then click **Properties**  on the toolbar to open its property sheet.
3. Click the **Format** tab, and in the **Picture** property box, type the path and file name for a .bmp, .ico, or .dib file. If you're not sure of the path or file name, click the **Build** button  to open the Picture Builder.

 [Tip](#)

In Microsoft Paint, you can create a file with a picture and caption for a button. Draw the picture and caption on a background of any size, and then select only the portion that will fit on the button. On the **Edit** menu, click **Copy To** and specify a file name in the **Copy To** dialog box.

 [Add a picture to a command button on a data access page](#)

1. Open a data access page in [Design view](#).
2. Click the command button you want to add the picture to, and then click **Properties**  on the toolbar.
3. Click the **Format** tab, and in the **BackgroundImage** property box, specify the location of image you want to use in the following format:

url("http://WebServerName/FileName")

4. Do one of the following:
 - To specify that only one image appears on the button, set the **BackgroundRepeat** property to **no-repeat**.
 - To specify that copies of the image appear in a horizontal line on the button, set the **BackgroundRepeat** property to **repeat-x**.
 - To specify that copies of the image appear in a vertical line on the

button, set the **BackgroundRepeat** property to **repeat-y**.

5. Set the position of the image in the **BackgroundPositionX** and **BackgroundPostionY** property boxes.

[Change the command button caption](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Click the label or button, and then select the label text or button text.



3. Type the new text.

[See the event procedure behind the command button](#)

The following procedure describes how to display an event procedure created by the Command Button Wizard on a form. In a data access page, you have to create the event procedure yourself.

1. In form Design view, right-click the command button created by the Command Button Wizard.
2. On the shortcut menu, click **Build Event**.

[Show All](#)

About command buttons

Command buttons provide you with a way of performing action(s) by simply clicking them. When you choose the button, it not only carries out the appropriate action, it also looks as if it's being pushed in and released.

[What are command buttons?](#)

You use a command button on a form or [data access page](#) to start an action or a set of actions. For example, you can create a command button that opens another form. To make a command button do something on a form, you write a [macro](#) or [event procedure](#) and attach it to the button's **OnClick** property. On a data access page, you can attach code written in either Microsoft JScript or Microsoft [Visual Basic Scripting Edition \(VBScript\)](#) to a command button by using the [Microsoft Script Editor](#).

You can display text or a picture on a command button in a form; you can display only text on a command button in a data access page.

[Creating command buttons](#)

You can create a command button on your own, or you can have Microsoft Access create your command button for you by using a wizard. A wizard speeds up the process of creating a command button because it does all the basic work for you. When you use a wizard, Access prompts you for information and then creates the command button based on your answers. By using the wizard, you can create more than 30 different types of command buttons. You can create command buttons to:

- Dial a phone number.
- Run a query or macro.
- Run or quit an application.
- Edit or apply a filter.
- Print or mail a report.
- Print the current record.
- Update form data.

- Find a specific record.

It's also a good idea to use the Command Button Wizard if you want to learn how to write event procedures. When Access creates a command button in a form or report with a wizard, it creates an event procedure and attaches it to the button. You can open the event procedure to see how it works and modify it to fit your needs. When you create a command button on a data access page with a wizard, Access doesn't create event procedures for you.

Wizards are not available in a stand-alone data access page or when you point a data access page to a database other than the one currently open. The Command Button Wizard is also not available if a data access page isn't bound to a table or query. You will have to create a command button on your own if wizards aren't available.

[Show All](#)

Create or modify a command button

Microsoft Access can create your [command button](#) for you if you use a wizard. However, wizards are not available in a stand-alone data access page, or when you point a data access page to a database other than the one that is currently open. The Command Button Wizard is also not available if the data access page isn't bound to a table or query. You will have to create a command button on your own if wizards aren't available.

[Create or modify a command button with a wizard](#)

1. Open a form or a data access page in [Design view](#).
2. Make sure the **Control Wizards**  tool in the toolbox is pressed in.
3. In the toolbox, click the **Command Button** tool.
4. On the form or data access page, click where you want to place the command button.
5. Follow the directions in the wizard dialog boxes. In the last dialog box, click **Finish** to display the command button in Design view.

[Create or modify a command button without a wizard](#)

1. Open a form or a data access page in [Design view](#).
2. Make sure the **Control Wizards**  tool in the toolbox is not pressed in. This turns off the wizard.
3. In the toolbox, click the **Command Button** tool .
4. On the form or data access page, click where you want to place the command button.
5. Make sure that the command button is selected, and then click **Properties** on the toolbar to open the command button's [property sheet](#).
6. In the **OnClick** property box on a form, enter the name of the macro or event procedure that you want to run when the button is clicked, or click the **Build** button to use the Macro Builder or Code Builder. On a data access page, you can attach code written in either [JScript](#) or Microsoft Visual Basic Scripting Edition ([VBScript](#)) to a command button by using the Microsoft Script Editor.

7. Do one of the following:

To display text on a command button on a data access page: Type the text in command button's **Value** property box.

To display text on a command button on a form: Type the text in the command button's **Caption** property box.

To display a picture on a command button on a form: In the command button's **Picture** property box, type the path and file name for a .bmp, .ico, or .dib file. If you're not sure of the path or file name, click the **Build** button  to open the Picture Builder.

 [Tip](#)

On a form, to Create or modify a command button that runs a macro, drag the macro from the Database window to form Design view.

 [Create a Cancel button](#)

1. Open a form in [Design view](#).
2. Click the command button, and then click **Properties**  on the toolbar to open the command button's property sheet.
3. In the **Cancel** property box, click **Yes**.

When a command button's **Cancel** property is set to **Yes** and the form is the active form, you can choose the command button by clicking it, pressing the ESC key, or pressing ENTER when the command button has the [focus](#). When the **Cancel** property is set to **Yes** for one command button, it's automatically set to **No** for all other command buttons on the form. Only one button can be the default on a form.

To have a **Cancel** button cancel all the actions that have taken place in a form or dialog box, you need to write a [macro](#) or [event procedure](#) and attach it to the **OnClick** property of the button.

 [Tip](#)

For a form that allows irreversible operations, such as deletions, it's a good idea

to make the **Cancel** button the default command button. To do this, set both the **Cancel** property and the **Default** property to **Yes**.



[See an event procedure created by the Command Button Wizard](#)

The following procedure describes how to display an event procedure created by the **Command Button Wizard** on a form. In a data access page, you have to create the event procedure yourself.

1. In form [Design view](#), right-click the command button created by the Command Button Wizard.
2. On the shortcut menu, click **Build Event**.

[Show All](#)

Create a script for a command button on a data access page

1. In [Design view](#) of a data access page, click **Microsoft Script Editor**  on the **Page Design** toolbar.
2. In the center pane of the Microsoft Script Editor, click the arrow next to the **Object** drop-down box on the left, and select the button that you want to create the procedure for.
3. Click the arrow in the **Event** drop-down box on the right, and select **onclick**.

[Show All](#)

Make a command button the default button in a form or custom dialog box

When you make a [command button](#) the default button on an active form, you can choose the command button by pressing ENTER (if no other command button has the focus) as well as by clicking the designated command button. When the **Default** property is set to **Yes** for one command button, it's automatically set to **No** for all other command buttons on the form. Only one button can be the default on a form.

1. Open a form in [Design view](#).
2. Click the command button, and then click **Properties** on the toolbar to open the command button's [property sheet](#).
3. In the **Default** property box, click **Yes**.



For a form that allows irreversible operations such as deletions, it's a good idea to make the **Cancel** button, if one exists, the default command button. To do this, set both the **Cancel** property and the **Default** property to **Yes**.

[Show All](#)

Make a command button the Cancel button

When the **Cancel** property of a command button on a form is set to **Yes** and the form is the active form, you can choose the command button by clicking it, pressing the ESC key, or pressing ENTER when the command button has the [focus](#). When the **Cancel** property is set to **Yes** for one command button, it's automatically set to **No** for all other command buttons on the form. Only one button can be the default on a form. This procedure does not apply to a command button on a [data access page](#).

1. Open a form in [Design view](#).
2. Click the command button, and then click **Properties**  on the **Form Design** toolbar to open the command button's property sheet.
3. In the **Cancel** property box, click **Yes**.

Note To have a **Cancel** button cancel all the actions that have taken place in a form or dialog box, you need to write a [macro](#) or [event procedure](#) and attach it to the **OnClick** property of the button.



For a form that allows irreversible operations, such as deletions, it's a good idea to make the **Cancel** button the default command button. To do this, set both the **Cancel** property and the **Default** property to **Yes**.

[Show All](#)

Customize a tab control

In form [Design view](#), do one or more of the following:

Add, delete, or change the order of tabs

- Right-click the border of the tab control, and then click **Insert Page**, **Delete Page**, or **Page Order**.

Change the tab order of controls on a page

- Right-click the page, and then click **Tab Order**.

Specify whether the tab control can have more than one row of tabs

- Double-click the edge of the tab control to display its property sheet, and then set the **MultiRow** property to **Yes**.

If the **MultiRow** property is set to **No**, Microsoft Access truncates tabs that don't fit within the width of the control and adds a scroll bar.

Specify whether to display tabs or command buttons at the top of the tab control

- Double-click the edge of the tab control to display its property sheet, and then set the **Style** property to **Tabs**, **Buttons**, or **None**.

You might want to click **None** if you plan to use a command button outside of the tab control to determine which page of the tab control has [focus](#).

Set the height of the tabs

- Double-click the edge of the tab control to display its property sheet, and then type a value in the **TabFixedHeight** box.

If the value is **0**, each tab will be high enough to fit its contents.

Set the width of the tabs

- Double-click the edge of the tab control to display its property sheet, and then type a value in the **TabFixedWidth** box.

If the value is **0**, each tab will be wide enough to fit its contents and, if there is more than one row of tabs, the width of each tab will increase so that each row of tabs spans the width of the tab control. If the value is greater than zero, all tabs have the identical width specified by this property.

Specify the font properties of the captions on tabs

- Double-click the edge of the tab control to display its property sheet, and then set the **FontName**, **FontSize**, **FontWeight**, **FontItalic**, or **FontUnderline** property.

You can't specify different settings for different tabs.

Specify the display text that appears on a tab

- Double-click the tab for the page to display its property sheet, and then type the text in the **Caption** box.

If you don't specify a caption, Microsoft Access uses the text in the **Name** property box.

Add a picture to a tab

- Double-click the tab for the page to display its property sheet, click the **Build** button (...) next to the **Picture** box, and then select a picture in the Picture Builder.

The picture appears to the left of the text specified in the **Caption** property. If you just want to display a picture and no text, enter a SPACEBAR space in the **Caption** property. Tab controls can contain bitmaps but not Microsoft Windows metafiles.

Note For more information about a property, click the property box in the property sheet and press F1.

[Show All](#)

About creating controls

[About bound, unbound, and calculated controls](#)

- A bound control is tied to a field in an underlying table or [query](#). In a [Microsoft Access project](#), a control can be bound to a field in a [view](#) or [stored procedure](#). If the Access project is connected to a database in Microsoft SQL Server 2000, a control can also be bound to a field in a [function](#). You use bound controls to display, enter, or update values from fields in your database.
- An unbound control doesn't have a data source. You can use unbound controls to display information, lines, rectangles, and pictures.
- A calculated control uses an [expression](#) as its source of data. An expression can use data from a field in an underlying table or query of a form or report, or from another control on the form or report.

The screenshot shows a form window titled "Employee Purchases". It contains three text boxes: "Product Name" with the value "Chang", "Unit Price" with the value "\$19.00", and "Employee Price" with the value "\$14.25". Below these is a text box containing the instruction "Please pay the accounting clerk." At the bottom of the form is a record navigation bar showing "Record: 14 | 2". Three blue numbered callouts (1, 2, 3) are placed on the form: callout 1 points to the Product Name and Unit Price fields; callout 2 points to the Employee Price field; callout 3 points to the text box below it.

- 1 These bound controls get their data from the ProductName and UnitPrice fields in the Products table.
- 2 This calculated control uses the following expression as its data source: = [UnitPrice]*.75. You specify the expression in the **ControlSource** property box.
- 3 Unbound controls such as this rectangle and freestanding label don't have data sources.

[About ways to create controls](#)

In a form or report that is based on a [record source](#), or in a [data access page](#) that is connected to a database, you can create a control by dragging a field from the [field list](#). Microsoft Access creates the appropriate control for the field and sets certain properties, based on the field's [data type](#).

If you want to specify the type of control you want, you can create the control by clicking the appropriate control tool in the toolbox. Then you can specify the properties of the control yourself, or you can use a wizard to create the control.

On forms and reports, you can use wizards to create command buttons, list boxes, [subforms](#), [combo boxes](#), and option groups. On data access pages, you can use wizards to create list boxes and [drop-down list boxes](#). Wizards are not available in a stand-alone data access page, or in a data access page that is connected to a database other than the one currently open.

[Show All](#)

About setting defaults for a control type

Each type of [control](#) that you can add to a form or report has a default set of properties that determines the general appearance and behavior of that type of control. For example, the default properties for a text box determine the size and font of the text within the box and whether the text box has an attached label. This set of properties for a control is called the "default control style." You may want to change the default control style for a type of control if you find that you often make the same changes to that type of control after you add it to a form or report.

For example, suppose you don't want to create labels for text boxes that you add to a new form. Rather than deleting the label each time you create a text box, you can change the **AutoLabel** property setting to **No** in the text box's default control style.

Once you reset a control's default control style, all subsequent controls of that type that you add to the current form or report will have those property settings. The default control style remains in effect for that form or report until you change it again. If you want the new settings to apply to all new controls, you can make the current form or report a template.

Note You can't set defaults for a control type that you add to a data access page.

[Show All](#)

Bind a control to a field

1. If the control is in a form or report, make sure that the form or report is based on a table or query. If the control is in a data access page, make sure that the data access page is connected to a database and that the section containing the control is bound to a [record source](#).
2. Open the form, report, or data access page in [Design view](#).
3. Make sure that the control is selected and then click **Properties**  on the toolbar to open its property sheet.
4. In the **ControlSource** property box, enter the name of the field that you want to bind the control to.

[Show All](#)

Create a bound span control

You can use a [bound span control](#) to display data from a field in the database or to display the result of an [expression](#). You can't create bound span controls on a form or report.

1. Open the data access page in [Design view](#).
2. Click the **Bound Span** tool  in the toolbox.
3. To bind the control to a field, in the [field list](#), click the field that you want to add to the page, and then drag it to the data access page.
4. In the data access page, release the field where you want the upper-left corner of the bound span control to be.

Notes

- If a bound span control is bound to a field that contains HTML text, set the **DataFormatAs** property of the control to **HTML** to display HTML-formatted text. The property is set to **TEXT** by default.
- Users cannot edit the contents of a bound span control in [Page view](#) or in Microsoft Internet Explorer.
- Use bound span controls instead of text boxes to improve the performance of a page.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Show All](#)

Set tab order

1. Open a form or data access page in [Design view](#).
2. Do one of the following:

[Change the tab order in a form](#)

1. On the **View** menu, click **Tab Order**.
2. Under **Section**, click the section you want to change.
3. Do one of the following:
 - If you want Microsoft Access to create a left-to-right and top-to-bottom tab order, click **Auto Order**.
 - If you want to create your own custom tab order, click the selector for the control you want to move. (Click and drag to select more than one control at a time.) Click the selector again and drag the control to the desired location in the list.
4. Click **OK**.

[Change the tab order in a data access page](#)

1. Select the [control](#) you want to move in the tab order, and then click **Properties**  on the **Page Design** toolbar to display the control's property sheet.
2. In the **TabIndex** property box, type the new tab order number.

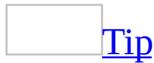
[Remove a control from the tab order](#)

1. Select the [control](#) you want to remove from the tab order, and then click **Properties**  on the **Form Design** or **Page Design** toolbar to display the control's property sheet.
2. Do one of the following:
 - If the control is in a form, in the **TabStop** property box, click **No**.

You can still click the control to select it as long as its **Enabled** property is set to **Yes**.

- If the control is in a data access page, set the **TabIndex** property to **-1**.

You can still click the control to select it as long as its **Disabled** property is set to **False**.



If you want to disable the control until a certain condition is met, you can use the **Enabled** property together with the **AfterUpdate** property.



You can change the tab behavior for the last field only in a form.

1. Double-click the [form selector](#) to open the form's property sheet.
2. In the **Cycle** property box, click one of the following settings.

Setting	Description
All Records	When you press TAB in the last field in the record, the focus moves to the first field in the next record.
Current Record	When you press TAB in the last field in the record, the focus moves back to the first field in the current record.
Current Page	When you press TAB in the last field in the form page, the focus moves back to the first field in the current page.

Note In Design view, the tab order is always the order in which you created the controls.

[Show All](#)

Assign an access key to a control

When you assign an [access key](#) to a label or button on a form or report, you can press ALT + an underlined character to move the [focus](#) to the control that is attached to that label, or to the button. In a [data access page](#), you assign the access key character to the control itself instead of to its label. You can then use ALT + that character to move the focus to that control.

1. Open a form, report, or data access page in [Design view](#).
2. Do one of the following:

Form or report

1. Select the label or button that you want to assign the access key to, and then click **Properties**  on the [toolbar](#) to open the control's property sheet.
2. In the **Caption** property box, type an ampersand (&) immediately before the character that you want to use as the access key.

Data access page

1. Select the control that you want to assign the access key to, and then click **Properties**  on the toolbar to open the control's property sheet.
2. In the **AccessKey** property box, type the character you want to use as the access key.

[Show All](#)

Add an ActiveX control (OLE custom control)

1. Open a form, report, or data access page in [Design view](#).
2. Click the **More Controls** tool  in the [toolbox](#), and then click the control you want in the list.

Note If you've installed [ActiveX controls](#) with other applications, such as Microsoft Excel, those controls will appear if you click the **More Controls** tool in the toolbox or if you click the **ActiveX Control** command on the **Insert** menu. Some ActiveX controls might not work well with Microsoft Access, however. To see a list of controls that have been tested and verified with Access, see the Readme file or contact the vendor for the control.

Security Use caution when you are adding ActiveX controls to your Web page. ActiveX controls may be designed in such a way that their use could pose a security risk. We recommend that you use controls from trusted sources only.

3. On the form, report, or data access page, click where you want to place the control.
4. Make sure the control is selected, and then click **Properties**  on the toolbar.
5. Set the properties you want.

To get Help on any property, click the property in the [property sheet](#), and then press F1.

[Show All](#)

HTML elements and classes for controls on a page

The following table shows the [HTML tags](#) and the [classes](#) that correspond to the [controls](#) you can add to a [data access page](#).

Control	HTML tags	Class
Text box	TEXTAREA	MsoTextbox
Bound span	SPAN	MsoBoundSpan
Label	SPAN	MSTheme-Label
Option group label	LEGEND	N/A
Scrolling text	MARQUEE	N/A
Option group	FIELDSET	N/A
Check box	INPUT (type=checkbox)	N/A
Option button	INPUT (type=radio)	N/A
List box	SELECT	N/A
Drop-down list box	SELECT	N/A
Command button	BUTTON	N/A
Expand	IMG	MsoExpandCollapse
Record navigation	TABLE	MsoNavContainer
Record navigation buttons	TD	MsoNavButton
Hyperlink	A	N/A
Image hyperlink	A IMG	N/A
Image	IMG	N/A
Movie	IMG	MsoMovie
Line	HR	N/A
Rectangle	SPAN	MsoRectangle

In the script for a data access page, [Microsoft Office Web components](#) are identified by OBJECT tags and by the component's unique class ID (classid) instead of by class.

[Show All](#)

Draw a rectangle or line

1. Open a form, report, or data access page in [Design view](#).
2. Do one of the following:

To draw a rectangle, click the **Rectangle** tool  in the [toolbox](#), and then click anywhere on the form, report, or data access page to create a default-sized rectangle. You can also drag to create a rectangle that is the size you want.

To draw a line, click the **Line** tool in the toolbox, and then click anywhere on the form, report, or data access page to create a default-sized line. You can click and drag to create a line that is the size you want.

Notes

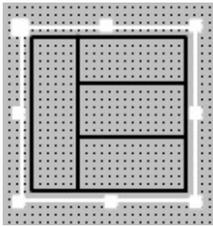
- To make small adjustments to the length or angle of a line in forms and reports, select the line, hold down the SHIFT key, and press one of the arrow keys.
 - To make small adjustments in the placement of a line, hold down the CTRL key and press one of the arrow keys.
3. To change the thickness of a rectangle's border or of a line, click the rectangle or line, click the arrow next to the **Line/Border Width**  button on the **Form/Report Formatting** toolbar or the **Page Formatting** toolbar, and then click the line thickness you want.
 4. To change the line style (dots, dashes, double, and so on) of a rectangle's border or of a line, click the rectangle or line, click **Properties**  on the toolbar to open the property sheet, and then click a border style in the **BorderStyle** property box.

Note If you place additional controls in a rectangle, select the rectangle and then click **Send to Back** on the **Format** menu to ensure that the controls and their associated screentips work properly.

[Show All](#)

Use a grid of grouped lines as a table for displaying data

1. In [Design view](#) of a form or report, create a grid of lines. For example, your grid might look like following:



2. Click outside of the grid of lines and drag a rectangle over the grid.
3. On the **Format** menu, click **Group**.

[Move a line in a grid of grouped lines](#)

1. Click the grid of grouped lines.
2. While holding down the ALT key, click the line you want to move. When the pointer changes to a double-headed arrow, you can drag the line to a new location.

[Rules for moving an interior vertical line](#)

- If you move the line to the right, the distance between the moving line and the nearest parallel line to the right remains constant, and the perpendicular line to the left of the moving line lengthens.
- If you move the line to the left, the distance between the moving line and the nearest parallel line to the right remains constant, and the perpendicular line to the left of the moving line contracts.

[Rules for moving an interior horizontal line](#)

- If you move the line upward, the distance between the moving line and the nearest parallel line below remains constant, and the perpendicular

line above the moving line contracts.

- If you move the line downward, the distance between the moving line and the nearest parallel line below remains constant, and the perpendicular line above the moving line lengthens.

[Rules for moving an exterior line](#)

- If you are moving the line away from the center of the grid, the nearest parallel line doesn't move, and the distance between the line you are moving and the nearest parallel line grows larger.
- If you are moving the line towards the center of the grid, the nearest parallel line doesn't move, and the distance between the line you are moving and the nearest parallel line grows smaller.

[Copy a line in a grid of grouped lines](#)

1. Select the line you want to copy and press CTRL+C.
2. Press CTRL+V to paste the line.
 - When you copy an interior vertical line, the new line is halfway between the line you copied and the line to the right of the line you copied.
 - When you copy an interior horizontal line, the new line is halfway between the line you copied and the line below the line you copied.
 - When you copy an exterior line, the new line is located halfway between the line you copied and the nearest line parallel to the line you copied.

Note After you paste the line, you do not need to recreate the group— the new line is automatically part of the group.

[Show All](#)

How control properties relate to properties in their underlying fields

When you create a bound [control](#) in a form, report, or data access page by dragging a field from the [field list](#), Microsoft Access copies certain properties from the field in the underlying table or [query](#) to the control. For example, if the Unit Price field in a Products table has the **Format** property set to **Currency**, when you create a bound text box in a form or data access page by dragging the Unit Price field from the field list, Access automatically sets the **Format** property for the text box control to the same value. (This assumes that the **DisplayControl** property for the field is set to **Text Box** or has no setting.)

It's best to set the **Format**, **DecimalPlaces**, **InputMask**, **ValidationRule**, **ValidationText**, and **DefaultValue** properties in the underlying field rather than in a control. This way, you can be sure that you have consistent settings whenever you add fields to a form or report.

 [Properties inherited for each type of bound control when you drag a field from the field list to a form or report](#)

Bound control	Properties inherited
Text box	Format , DecimalPlaces , InputMask , Caption , and StatusBarText (from Description property)
List box	All properties specified in the Lookup tab in table Design view , Caption property and StatusBarText (from Description property)
Combo box	All properties specified in the Lookup tab in table Design view, Format , InputMask , and Caption properties, StatusBarText (from Description property)
Check box, option button, option group, toggle button, bound object frame	Caption property and StatusBarText (from Description property)

If you open the form in [PivotTable view](#) or [PivotChart view](#), a bound text box inherits the **Format**, **DecimalPlaces** and **Caption** properties. Other bound controls inherit the **Caption** property only.

Note In a [Microsoft Access database](#), the **DefaultValue**, **ValidationRule**, and **ValidationText** properties are not inherited by controls in forms. That is, Access doesn't set the control properties to the values that are set for the same properties in the underlying field. These property settings are enforced, however, when you create a control that is based on a field with these settings. If you set the **ValidationRule** property in a field, and also in a control based on that field, Access enforces both rules. If you set the **DefaultValue** property for a field, and also in a control that is based on that field, the control property will override the field property.

 [Properties inherited by bound controls when you drag a field from the field list in a data access page](#)

In a data access page, a text box, list box, or drop-down list box can inherit **Format**, **Caption**, and **DefaultValue** properties.

A list box or drop-down list box can inherit **RowSource** and **ColumnCount** properties specified in the **Lookup** tab in table [Design view](#) if the **RowSourceType** property is set to **Table/Query** in a [Microsoft Access database](#) or **Tables/Views/Functions** in an [Access project](#). If the **RowSourceType** property of the field is set to **Value List** or **Field List**, the control doesn't inherit these properties. In addition, the control doesn't inherit the **ColumnCount** property if the value of the property is greater than 2.

Note In a form, if you create a text box that is based on a field with the [Memo data type](#), Access automatically sets the **EnterKeyBehavior** property to **New Line In Field**, and the **ScrollBars** property to **Vertical Only**. In a data access page, if you create a text box based on a Memo field, Access automatically sets the **Overflow** property to **Auto**. These properties aren't inherited from the field properties—they are properties of the control.

You can change the settings of any inherited (copied) properties in the control's property sheet. If you change a control's property setting, the change won't affect the setting of that property for the field in the underlying table or query.

Changing the inherited property of a field that has form and report controls bound to it

If you change an inherited property of a field that has controls bound to it, Access gives you an option to automatically update all the controls that inherit the property setting. For example, the Unit Price field in the Product table is the source for several textbox controls that appear on different forms and reports in the Sales database. The **Format** property of these controls inherit the **Format** property setting of the Unit Price field. When you update the **Format** property of the Unit Price field in Design view, Access displays the **Property Update Options** button next to the property. Clicking the **Update...** option on the drop-down menu of the button displays the **Update Properties** dialog box.



The dialog box lists the objects that contain controls that inherit the field property setting. You can choose to update all, some, or none of the objects. Access might not succeed in updating the controls, if you do not have exclusive access to the database.

The update functionality is not available in Access projects. In Access databases, the **Property Update Options** button does not update controls on a data access page. If the updated field is in a linked table, Access will not update the objects in the source database.

You can turn off the display of the **Property Update Options** button by clearing the **Show Property Update Options buttons** check box in the **Options** dialog box.

Note If you change the field property settings for the **DefaultValue**, **ValidationRule**, and **ValidationText** properties, the changes will always be enforced in a form for controls based on these fields, irrespective of whether the controls were created before or after you changed the field properties.

Show or hide the Property Update Options button

1. On the **Tools** menu, click **Options**.
2. On the **Tables/Queries** tab, select or clear the **Show Property Update Options buttons** check box.

Propagate changes in field properties to bound controls

When you make a change to a field's inherited property in Table design view, you can choose to update the corresponding property of controls on forms and reports that are bound to the field. If a bound control does not inherit the field's property value, and has a different value specified in its property sheet, Access does not update the control's property.

1. Open the table that contains the field in Design view.
2. Under **Field Name**, click the field whose property you want to change.
3. Change the description of the field, or one of the properties in the **General** or **Lookup** tab, and press Enter.

If you changed the value of an inherited property, you will see the **Property Update Options** button next to the property name.

4. Click the **Property Update Options** button, then click the **Update...** command.
5. In the **Update Properties** dialog box, select the forms and reports that contain the controls that must be updated, and click **Yes**.

Notes

- This functionality is not available in Access projects. In Access databases, the **Property Update Options** button does not update controls on a data access page. If the updated field is in a linked table, Access will not update the objects in the source database.
- You can turn off the display of the **Property Update Options** button by clearing the **Show Property Update Options buttons** check box in the **Options** dialog box.

[Show All](#)

About customizing a control

In [Design view](#) of a form, report, or [data access page](#), you can format a control in the following ways:

Size. You can resize a control by using the sizing handles, or you can resize a control to the height or width of another control. When you are in Design view of a form or report, you can use the ruler to help you resize controls. The width of the ruler's unit of measure is based on the regional settings in Microsoft Windows Control Panel.

Color, transparency, and special effects. You can change the color of a control or make it transparent. You can also emphasize the appearance of a control by using special effects. For example, you can specify that a control is raised, sunken, or etched. On a form or report, you can also specify that a control is shadowed or chiseled.

Borders. You can set the color and width of borders of controls. Also, you can set the line style of borders. For example, you can specify that a border consists of dashes or dots.

Appearance of text. You can specify the font and font size for text in a control. Also, you can make the text in a control bold, italic, or underlined.

Position. You can move a control or align it relative to another control. You can increase or decrease the space between controls, or you can specify that controls are evenly spaced.

About changing the properties of several controls at once

You can change property settings for a group of controls of the same type or for a group of controls of different types. If you select controls of different types, Microsoft Access displays only the properties that are shared by the group in the property sheet. If all the controls that you've selected share the same property setting, that setting appears in the property sheet; otherwise, that property box is blank. If you change the property setting, the change is applied to all selected controls.

Notes

- To select several controls at once in a data access page, you must have Microsoft Internet Explorer 5.5 or later installed on your computer.
- In a form or report, the **Name** property never appears when more than one control is selected because control names must be unique.

[Show All](#)

About display formats for fields or controls (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can use the **Format** property to specify the data [display format](#) for a field or a [control](#). For example, you can choose to have all dates formatted using the month/day/year format or some other format. You can choose from a list of predefined formats for fields with the AutoNumber, Number, Currency, Date/Time, and Yes/No [data types](#), or you can create your own custom format for any field data type other than OLE Object.

In most cases, you'll want to set the property for the field in table [Design view](#). That way, the setting is automatically applied to the field in query Design view and to controls on a form or report [bound](#) to that field. In certain cases, however, you may want to set the property in query Design view, form Design view, or report Design view. For example, you may want to display a different format in a table datasheet than in a report. You could set the **Format** property for the field in query Design view and then base your report on the query. If you're using an [unbound control](#), you'll need to set the **Format** property in form Design view or report Design view.

If you want to create literal display characters that help you enter data in a predefined format, you may want to use an [input mask](#) instead of a format.



[Choosing between a display format and an input mask](#)

Microsoft Access provides two field properties that produce similar results: the **Format** property and the **InputMask** property.

Use the **Format** property to display data in a consistent format. For example, if you set the **Format** property for a Date/Time field to **Medium Date** format, all dates entered will display in this form: 12-Jan-1999. If a user of your database enters a date as 01/12/99 (or any other valid date format), Microsoft Access will convert the display to the **Medium Date** format when the record is saved.

The **Format** property affects only how a value is displayed, not how it is stored in the table. Also, a [display format](#) isn't applied until the data entered is saved — nothing is displayed in the field to suggest or control the format in which data is entered. If you want data to display exactly as entered, don't set the **Format** property. If you need to control how data is entered, use an [input mask](#) in addition to, or instead of, a data display format. An input mask ensures that the data will fit in the format you define, and you can specify the kind of values that can be entered.

If you define both a display format and an input mask for a field, Microsoft Access uses the input mask when you are adding or editing data, and the **Format** setting determines how the data is displayed when the record is saved. When using both **Format** and **InputMask** properties, be careful that their results don't conflict.

[International data display formats](#)

Microsoft Access can display a wide range of international formats. To help ensure consistency among applications, Microsoft Access uses the regional settings in Microsoft Windows Control Panel for the predefined Number and Date/Time formats. If your data isn't displayed in the right format for your country/region, change the regional settings specified in Control Panel.

Changes made to the formats of regional settings in Windows Control Panel will be automatically reflected in your database for Currency fields if the **Format** property setting is blank. However, this only affects the format; no conversion of currency values will be made. Therefore, transferring data between computers with different regional settings can result in incorrect currency data.

For example, when using the Currency format on a field where the **Format** property setting is blank, a value of 5,47 kr on a computer set for Denmark converts to \$5.47 on a computer set for the United States. To prevent such errors, define a custom format for the currency, such as #,## kr. The custom format overrides the regional settings specified in Windows Control Panel. Similar problems won't occur when transferring standard number, date, or time data between computers with different regional settings.

[Show All](#)

Add or remove scroll bars

[Add or remove scroll bars for a form or for a text box on a form](#)

By default, a form has vertical and horizontal scroll bars that appear only when necessary. Also, by default, when you create a text box based on a [Memo](#) field in a [Microsoft Access database](#), or when you create a text box based on a [text](#) or [ntext](#) column in an [Access project](#), Access adds a vertical scroll bar to the text box. Otherwise, a scroll bar doesn't appear in a text box unless you change the value of the **ScrollBars** property. Follow this procedure only if you want to change the default behavior.

1. Open the form in [Design view](#).
2. Do one of the following:
 - To open the property sheet for a text box, select the text box and then click **Properties**  on the toolbar.
 - To open the property sheet for a form, double-click the [form selector](#).
3. Click the **Format** tab, and in the **ScrollBars** property box, do one of the following:
 - To specify that a scroll bar for a text box never appears, click **None**.
 - To specify that a scroll bar for a form never appears, click **Neither**.
 - To specify that a vertical scroll bar for a text box always appears, click **Vertical**.
 - To specify that a vertical scroll bar for a form appears when necessary, click **Vertical Only**.
 - To specify that a horizontal scroll bar for a form appears when necessary, click **Horizontal Only**.
 - To specify that both vertical and horizontal scroll bars appear when necessary, click **Both**.

[Add or remove scroll bars for a text box on a data access page](#)

By default, scroll bars for a text box based on a Memo field appear only when necessary; if the text box is based on a field whose data type isn't Memo, the scroll bars do not appear at all. Follow this procedure only if you want to change the default behavior.

1. Open the data access page in [Design view](#).
2. Click the text box, and then click **Properties**  on the toolbar.
3. Click the **Format** tab, and do one of the following:
 - To specify that scroll bars never appear in the text box, set the **Overflow**, **OverflowX**, and **OverflowY** properties to **Hidden**.
 - To specify that scroll bars appear in the text box only when necessary, set the **Overflow**, **OverflowX**, and **OverflowY** properties to **Auto**.
 - To add a vertical scroll bar to a text box, set the **OverflowX** property to **Hidden** and the **OverflowY** property to **Scroll**.
 - To add a horizontal scroll bar to a text box, set the **OverflowX** property to **Scroll** and the **OverflowY** property to **Hidden**.

[Show All](#)

Align controls to each other

[Align controls to each other on a form or report](#)

1. Open a form or report in [Design view](#).
2. To select the [controls](#) that you want to align, hold down the SHIFT key and click each control, or click the **Select Objects** tool  in the toolbox and then click and drag a rectangle over the controls.

Select only controls that are in the same row or in the same column.

3. On the **Format** menu, point to **Align**, and then click one of the following commands:
 - **Left** Aligns the left edges of the controls with the left edge of the leftmost control.
 - **Right** Aligns the right edges of the controls with the right edge of the rightmost control.
 - **Top** Aligns the top edges of the controls with the top edge of the highest control.
 - **Bottom** Aligns the bottom edges of the controls with the bottom edge of the lowest control.

Note If any of the selected controls would overlap after being aligned, Microsoft Access places them with their edges next to each other instead.

[Align controls to each other in a data access page](#)

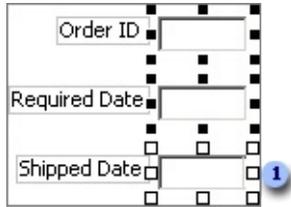
In a data access page, how you align controls to each other depends on whether you have Microsoft Internet Explorer 5.5 or 5 installed.

- Open the data access page in [Design view](#), and do one of the following.

[If you have Internet Explorer 5.5 installed](#)

1. Select the [controls](#) that you want to align by holding down the SHIFT key and clicking each control.

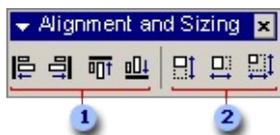
Make sure that the control that you want to align the other controls to is the last control you select and that its sizing handles are white. This control is the primary selection.



- 1 The sizing handles of the primary selection are white.
2. On the **Format** menu, point to **Align**, and then click one of the following commands:
 - **Left** Aligns the left edges of the controls with the left edge of the primary selection.
 - **Right** Aligns the right edges of the controls with the right edge of the primary selection.
 - **Top** Aligns the top edges of the controls with the top edge of the primary selection.
 - **Bottom** Aligns the bottom edges of the controls with the bottom edge of the primary selection.
 - **Align Middle** Aligns the controls so that they are centered horizontally relative to the primary selection.
 - **Align Center** Aligns the controls so that they are centered vertically relative to the primary selection.

[If you have Internet Explorer 5 installed](#)

1. On the **Format** menu, click **Alignment and Sizing** to display the **Alignment and Sizing** toolbar.



- 1 **Align Left**, **Align Right**, **Align Top**, and **Align Bottom** buttons
- 2 **Size Height**, **Size Width**, and **Size Height and Width** buttons

2. Make sure the [control](#) that is aligned the way you want is selected.
3. Do one of the following:
 - To align just one control, click the button that you want on the **Alignment and Sizing** toolbar, and then click the control that you want to align to the selected control.
 - To align several controls, lock the button that you want on the **Alignment and Sizing** toolbar by double-clicking it, and then click the controls that you want to align to the selected control.
 - To release the lock, click the same or a different button on the **Alignment and Sizing** toolbar or press ESC.

[Align controls to each other by using the grid](#)

1. Open a form, report, or data access page in [Design view](#).
2. If the grid isn't already displayed, click **Grid** on the **View** menu.
3. Select the [control](#) or controls that you want to align. To select more than one control, hold down the SHIFT key and click each control, or drag a rectangle over the controls.

Note To select more than one control in a data access page, you must have Microsoft Internet Explorer 5.5 installed.

4. On the **Format** menu, point to **Align**, and then click **To Grid**.

[Show All](#)

Change the appearance of the border of a control

1. Open a form, report, or data access page in [Design view](#).
2. Click the [control](#) you want to change.
3. Do one of the following:

[Change the color of a border](#)

1. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to **Line/Border Color** .
2. Click a color in the palette.

[Tip](#)

If you want to use the same color for another control, you can just click **Line/Border Color**  (you don't have to click the arrow next to the button and pick the color again from the palette). If you want to set different colors for several controls at once, click the arrow next to **Line/Border Color** , click the horizontal bar at the top of the color palette, and drag it to another location. This way, you don't need to click **Line/Border Color**  each time you want to apply a color.

[Change the width of a border](#)

1. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to the **Line/Border Width**  button.
2. Click the border width you want.

[Tip](#)

If you want to use the same border width for another control, you can just click the **Line/Border Width**  button (you don't have to click the arrow next to the button and pick the width again from the list).

[Make a border transparent](#)

1. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to **Line/Border Color** .
2. Click the **Transparent** button on the palette.

[Make a border dotted, dashed, or solid](#)

1. Select the control, and then click **Properties**  on the **Form Design**, **Report Design** or **Page Design** toolbar to open its property sheet.
2. Do one of the following:

In a form or report In the **BorderStyle** property box, click the border style you want to apply.

In a data access page In the **BorderStyle** property box, type **dotted**, **dashed**, or **solid**.

Note In a data access page, you can't make a border dotted or dashed unless you have Microsoft Internet Explorer 5.5 or later installed.

Note In forms and reports, if the **SpecialEffect** property for the control is set to **Raised**, **Sunken**, **Etched**, or **Chiseled**, Microsoft Access changes the setting to **Flat** when you change the color, width or style of a border or make a border transparent.

[Show All](#)

Change a control to another control type

1. Open a form or report in [Design view](#).
2. Click the [control](#) that you want to change.
3. On the **Format** menu, point to **Change To**.

Microsoft Access displays a list of controls that the selected control can be changed to. If a type of control appears dimmed on the menu, you can't change the selected control to that type of control.

4. Click the type of control that you want to change to.

When you change a control to another type of control, Access copies the appropriate property settings from the original control to the new control. If a property exists for the original control but not for the new control, Access doesn't copy that setting. If a property exists for the new control but not for the original control, Access sets the property to the [default control style](#) for that type of control.

Note You can't change a control to another type of control in a [data access page](#).

[Show All](#)

Show or hide a control

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#) you want to show or hide, and then click **Properties**  on the **Form Design**, **Report Design** or **Page Design** toolbar to open its property sheet.
3. Do one of the following:

To show or hide a control in a form or report

- In the **Visible** property box, click **Yes** or **No**.

To show or hide a control in a data access page

- In the **Visibility** property box, click **Visible** or **Hidden**. If you want the control to inherit the **Visibility** property of the section where it is located, click **Inherit**.
- If the control has an associated label, you must set the label's **Visibility** property separately.

Note To specify whether a column is visible in [Datasheet view](#) of a form, use the **ColumnHidden** property.

[Tips](#)

- On forms and reports, to hide or unhide a control in response to an event, you can use a [macro](#) or [event procedure](#) to set the **Visible** property. Attach the macro or event procedure to the **OnCurrent** property of a form or the **OnFormat** property of a report. For example, you can show or hide a congratulatory message next to a salesperson's monthly sales total in a sales report, depending on the sales total.
- To hide a section or control on a form only when you are printing, use the **DisplayWhen** property.

[Show All](#)

Make a control transparent

When a [control](#) is transparent, the color displayed in the control is the color of the control, [section](#), or form behind it.

1. Open a form, report, or data access page in [Design view](#).
2. Click the control that you want to change.
3. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to **Fill/Back Color** .
4. Click the **Transparent** button on the palette.

Note You can make any control transparent except labels, list boxes, and toggle buttons in forms and reports. In data access pages, you can make drop-down list boxes, list boxes, rectangles, and PivotTable lists transparent. Option groups in forms, reports, and data access pages are transparent by default.

[Show All](#)

Show a tip when the pointer moves over a control

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#) that you want to create a tip for, and then click **Properties**  on the **Form Design, Report Design, or Page Design** toolbar to open the control's property sheet.
3. In the **ControlTipText** (forms and reports) or the **Title** (data access pages) property box in the property sheet, type the message you want to display for the control. You can type up to 255 characters.

[Show All](#)

Show instructions for a control in the status bar

1. Open a form or report in [Design view](#).
2. Make sure that the [control](#) is selected, and then click **Properties**  on the **Form Design** or **Report Design** toolbar to open its property sheet.
3. In the **StatusBarText** property box, type the message that you want to display for the control.

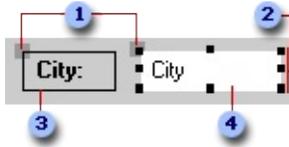
You can enter up to 255 characters, but Microsoft Access can only display as much text as will fit in the status bar. The amount of text that can be displayed varies according to the font you use and whether you resize the Access window.

Note This procedure does not apply to [data access pages](#).

[Show All](#)

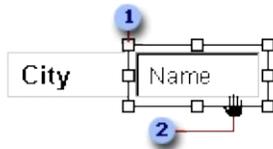
Select one or more controls

In a form or report, if the [control](#) has an attached [label](#), it looks like this:



- 1 Move handles
- 2 Sizing handles
- 3 Attached label
- 4 Selected control

In a [data access page](#), if the control has an attached label, it looks like this:



- 1 Move handles
 - 2 When the pointer changes to an open hand, you can drag the control.
1. Open a form, report, or data access page in Design view.
 2. Do one of the following:
 - o To select one control, click anywhere in the control.
 - o To select adjacent controls, click at any point outside a control, and drag a rectangle over the controls that you want to select.
 - o To select nonadjacent or overlapping controls, hold down the SHIFT key and click each control that you want to select.

Note In a data access page, you can't select more than one control unless you have installed Microsoft Internet Explorer 5.5 or later.

3. If you want to remove one or more controls from the selected group, hold down the SHIFT key, and then click the controls that you want to remove.

Note If you click a control that's already selected, an insertion point appears inside the control so that you can enter or edit text. To reselect the control, click outside the control, and then click the control again.



You can specify whether the rectangle that you drag over controls needs to only touch the controls or must enclose them completely to select them. On the **Tools** menu, click **Options**, click the **Forms/Reports** tab, and then click one of the options under **Selection Behavior**.

[Show All](#)

Customize the appearance of text in a control

Customize the appearance of text in an existing control

1. Open a form, report, or data access page in [Design view](#).
2. Click the [control](#) that contains the text you want to change.
3. Do one or more of the following:

[Change the alignment of text](#)

- On the **Form/Report Formatting** or **Page Formatting** toolbar, click **Center**  to center the text, **Align Right**  to align the text to the right, or **Align Left**  to align the text to the left.

Note If you don't select any of the alignment buttons, text is aligned to the left and numbers are aligned to the right.

[Change the color of text](#)

1. On the **Form/Report Formatting** or **Page Formatting** toolbar, click the arrow next to **Font/Fore Color** .
2. Click a color in the palette.

[Tip](#)

If you want to use the same color for another control, you can just click **Font/Fore Color**  (you don't have to click the arrow next to the button and pick the color again from the palette). If you want to set different colors for several controls at once, click the arrow next to **Font/Fore Color** , and then click the horizontal bar at the top of the color palette and drag it to another location. By using this method, you won't need to click **Font/Fore Color**  each time you want to apply a color.

 [Change the font or font size, or make text bold, italic, or underlined](#)

On data access page [scrolling text controls](#), you can select text and apply a different font size or font style such as bold or italic to individual characters within the control. In all other controls, when you change the font size, the change applies to the entire contents of the control.

- Do one or more of the following:

Change the font

- In the **Font Name** box  on the **Form/Report Formatting** or **Page Formatting** toolbar, click the font you want to apply.

Change the font size

- In the **Font Size** box  on the **Form/Report Formatting** or **Page Formatting** toolbar, enter the font size that you want to use.

Make text bold, italic, or underlined

- On the **Form/Report Formatting** or **Page Formatting** toolbar, click **Bold**  to make the text bold, **Italic**  to make the text italic, or **Underline**  to make the text underlined.

 [Tip](#)

If you want to resize the control to fit the new font, point to **Size** on the **Format** menu, and then click **To Fit**.

Change the alignment of text in new controls or in labels attached to new controls

Changes made by using this procedure only affect new controls on the current form or report. This procedure doesn't apply to [data access pages](#).

1. Open a form or report in Design view.
2. Do one of the following:

[Change the alignment of text in new controls](#)

1. In the [toolbox](#), click the tool for the type of control whose default alignment you want to change. For example, if you want to change the default alignment of text in a text box, click the **Text Box** tool  in the toolbox.
2. Click **Properties**  on the toolbar to display the property sheet.

Microsoft Access changes the title bar of the property sheet to indicate that you're setting default properties for that type of control.

3. In the **TextAlign** property box, click **General**, **Left**, **Center**, **Right**, or **Distribute**. If you click **General**, Access aligns text to the left and numbers and dates to the right. For more information, click in the **TextAlign** property box and press F1.

[Change the alignment of text in labels attached to new controls](#)

1. Click the tool in the [toolbox](#) for the type of control that has the attached label. For example, if you want to change the default alignment of text in labels that are attached to text boxes, or if you want to change the default alignment of text in a text box or in a label, click the **Text Box** tool  in the toolbox.
2. Click **Properties**  on the toolbar to display the property sheet.

Microsoft Access changes the title bar of the property sheet to indicate that you're setting default properties for that type of control.

3. In the **LabelAlign** property box, click **Left**, **Center**, **Right**, or **Distribute**.

For information about how the **LabelAlign** property interacts with the **LabelX** and **LabelY** properties, click the **LabelX** or **LabelY** property box and press F1.

[Show All](#)

Move one or more controls

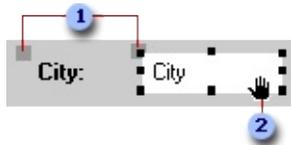
1. Open a form, report, or [data access page](#) in [Design view](#).
2. Do one of the following:

[Move a control and its label together](#)

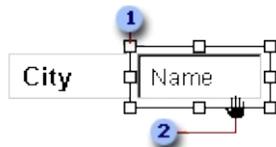
1. Click the [control](#) or its [label](#).

When you click either part of a [compound control](#) in forms and reports, Microsoft Access displays the move handles for both controls, as well as the sizing handles for the control that you clicked. On data access pages, when the control is selected, the label is not automatically selected. If you move the control, the label goes with it. But if you move the label, the control does not move.

2. Move the pointer over the border (not the move handle) of either the control or its label. After the pointer changes to an open hand, you can drag the control or controls to a new position.



- 1 Move handles of a control in a form or report.
- 2 When the pointer changes to an open hand, you can drag the control.



- 1 Move handle of a control in a data access page
- 2 When the pointer changes to an open hand, you can drag the control.

[Move a control and its label separately](#)

In a data access page, you can't move a control separately from its label, but you can select the label and move it separately from the control.

1. Click the [control](#) or its [label](#).
2. Do one of the following:
 - In a form or report, place the pointer over the move handle in the upper-left corner of the control or label, and the pointer changes to an upward-pointing hand.



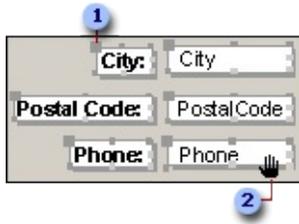
- 1 When the pointer changes to an upward-pointing hand, you can drag the control.
 - In a data access page, place the pointer on any edge of the label, and the pointer changes to an open hand.
3. Drag the control or label to its new position.

Notes

- The parts of a [compound control](#) remain related, even if you move the individual parts separately.
- To move an attached label to another [section](#) without moving the control, you must use the **Cut** and **Paste** commands. In a form or report, if you move a label to another section, the label will no longer be related to the control.

[Move two or more controls at the same time](#)

1. Select the controls by holding down the SHIFT key as you click each control.
2. Pass the pointer over the border (not the move handle) of any selected control until it turns into an open hand, and then drag the controls to a new position.



- 1 Move handle
- 2 When the pointer changes to an open hand, you can drag the controls.



[Move a control in front of or behind other controls](#)

- Click the [control](#) that you want to move, and then on the **Format** menu, click **Send to Back** or **Bring to Front**.

Note A list box or [subform](#) is always in front of other controls; you can't format a list box or subform so that it is behind another control.



[Tips](#)

- When you move a control in a form or report, you can maintain its horizontal or vertical alignment with other controls by holding down the SHIFT key while selecting and moving the control. You can move the control horizontally or vertically only, depending on which direction you move it first.
- If you need to make small adjustments to the placement of a control, you might find it easier to press the appropriate arrow keys. If **Snap to Grid** is on when you use the arrow keys to move a control, Access aligns the control to the grid.
- If you want to temporarily override the current **Snap to Grid** setting, hold down the CTRL key while you're placing, moving, or sizing a control. For example, if **Snap to Grid** is turned on, you can hold down the CTRL key and move a control anywhere on the form, report or data access page without aligning it to the grid. Releasing the CTRL key restores the current **Snap to Grid** setting.

[Show All](#)

Make a control appear raised, sunken, shadowed, chiseled, or etched

1. Open a form, report, or data access page in [Design view](#).
2. Click the [control](#) that you want to change.
3. On the **Form/Report Formatting** or the **Page Formatting** toolbar, click the arrow next to the **Special Effect**  button.
4. Click the effect you want to apply.

Note In forms and reports, if you click **Raised**, **Sunken**, **Etched**, or **Chiseled**, Microsoft Access overrides any settings for the **BorderColor**, **BorderWidth**, and **BorderStyle** properties. If you click **Etched**, Access also sets the **BackStyle** property for text boxes and combo boxes to **Transparent**, and the **BackColor** property for list boxes to the same setting as the section's **BackColor** property.

[Tips](#)

- If you want to use the same special effect for another control, you can just click the **Special Effect**  button (you don't have to click the arrow next to the button and pick the effect again from the list).
- In a form or report, you can adjust the thickness of a control's shadow by clicking the **Line/Border Width**  button on the **Form/Report Formatting** toolbar.

[Show All](#)

Make a control read-only

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Make sure that the [control](#) is selected, and then click **Properties**  on the **Form Design, Report Design, or Page Design** toolbar to open its property sheet.
3. In a form or report, do one of the following:
 - If you want to disable the control completely so that it's dimmed and can't receive the [focus](#), set the **Enabled** property to **No**.
 - If you want to make data in the control readable, but not allow users to change the data, set the **Locked** property to **Yes**. If you set the **Enabled** property to **No** and the **Locked** property to **Yes**, the control won't appear dimmed, but it won't be able to receive the focus.

In a data access page, to disable the control completely so that it's dimmed and can't receive the focus, set the **Disabled** property to **True**.



In a form or report, you can use the **Enabled** property together with the **AfterUpdate** property to disable a control until a user enters text or makes a selection in another control. For example, you might want to disable a command button until data has been entered in a text box. You can use the **AfterUpdate** property of the text box to call an [event procedure](#) or a macro to enable the command button.

[Show All](#)

Change default properties for a control type

In a form or report, you can change the default properties for new controls. This procedure does not apply to data access pages.

1. Open a form or report in [Design view](#).
2. Do one of the following:

[Change the default settings for a type of control](#)

1. In the [toolbox](#), click the tool for the type of control that you want to change. For example, if you want to change the default control style for text boxes, click the **Text Box** tool .
2. Display the property sheet by clicking **Properties**  on the **Form Design** or **Report Design** toolbar.

Microsoft Access changes the title bar of the property sheet to indicate that you've opened the default property sheet for that type of control.

3. In the property sheet, change the property settings.
4. If you want the new settings to apply to all new controls, make the current form or report a template.

[How?](#)

You can either use an existing form or report as a template, or create a form or report specifically for use as a template. Changing the default template does not have any effect on existing forms and reports.

1. On the **Tools** menu, click **Options**.
2. Click the **Forms/Reports** tab.
3. Type the name of the new template in the **Form Template** or **Report Template** box.

4. Click **OK**.

Note To see the name of the template that is currently used for new forms or reports, click **Options** on the **Tools** menu, and then click the **Forms/Reports** tab.

[Change the default settings for a type of control to match the settings of an existing control](#)

1. Select the control that has the characteristics that you want to use as defaults.
2. On the **Format** menu, click **Set Control Defaults**.

Microsoft Access sets the default properties for that type of control based on the control you selected.

[Show All](#)

Change the spacing between controls

On a data access page, to select multiple controls and adjust the space between them, you must have Microsoft Internet Explorer 5.5 or later installed.

1. Open a form, report, or data access page in [Design view](#).
2. Select the [controls](#) that you want to adjust by holding down the SHIFT key and clicking each control. For controls with attached labels, select the controls, not their labels. If you want to space controls evenly, select at least three controls.
3. Do one of the following:

Space controls evenly

- On the **Format** menu, point to **Horizontal Spacing** or **Vertical Spacing**, and then click **Make Equal**.

Microsoft Access spaces the controls evenly. Only the middle control or controls actually adjust, however; the outside controls don't change location.

Increase or decrease space

- On the **Format** menu, point to **Horizontal Spacing** or **Vertical Spacing**, and then click **Increase** or **Decrease**.

When the space between controls is increased or decreased, the leftmost (horizontal spacing) and highest (vertical spacing) controls don't change positions.

Remove the space between controls on a data access page

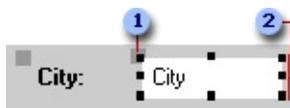
- On the **Format** menu, point to **Horizontal Spacing** or **Vertical Spacing**, and then click **Remove**. The leftmost or top control remains in place.

[Show All](#)

Resize a control

[Resize a control](#)

1. Open a form, report, or [data access page](#) in [Design view](#).
2. Select the [control](#) or controls that you want to resize. To select more than one control, hold down the SHIFT key and click each control, or drag a rectangle over the controls.
3. Drag the sizing handles until the control is the size you want.



- 1 Move handle
- 2 Sizing handles

If you select more than one control, all the controls change size as you drag the sizing handle of one control.

Note In data access pages, you can't resize a [drop-down list box](#) vertically. The vertical size of the drop-down list box is based on the font setting of the data in the control.

[Tip](#)

If you want to make small adjustments to the size of a control in a form or report, you might find it easier to press SHIFT plus the appropriate arrow keys.

[Size controls uniformly relative to each other on a form or report](#)

1. Open a form or report in [Design view](#).
2. Select the [controls](#) that you want to size by holding down the SHIFT key and clicking each control, or by dragging a rectangle over the controls.
3. On the **Format** menu, point to **Size**, and then click one of the following commands:

- **To Tallest** Makes the controls as tall as the tallest selected control.
- **To Shortest** Makes the controls as short as the shortest selected control.
- **To Widest** Makes the controls as wide as the widest selected control.
- **To Narrowest** Makes the controls as narrow as the narrowest selected control.

[Size controls uniformly relative to each other on a data access page](#)

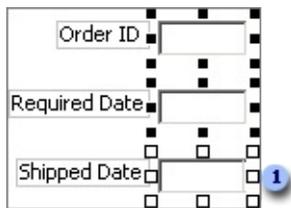
How you size controls relative to each other depends on whether you have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2), or 5.5 or later installed.

- Open a data access page in [Design view](#).

[If you have Internet Explorer 5.5 or later installed](#)

1. Select the [controls](#) that you want to resize by holding down the SHIFT key and clicking each control.

Make sure that the control that you want to align the other controls to is the last control you select and that its sizing handles are white.

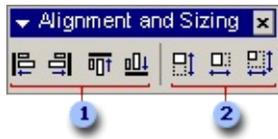


- 1 The sizing handles of the primary selection are white.
2. On the **Format** menu, point to **Size**, and then click one of the following commands:
 - **Height** Makes the controls as tall as the primary selection.
 - **Width** Makes the controls as wide as the primary selection.
 - **Both** Makes the controls as tall and as wide as the primary selection.

[If you have Microsoft Internet Explorer 5.01 with Service Pack 2](#)

[\(SP2\) installed](#)

1. On the **Format** menu, click **Alignment and Sizing** to display the **Alignment and Sizing** toolbar.



- 1 **Align Left, Align Right, Align Top, and Align Bottom** buttons
 - 2 **Size Height, Size Width, and Size Height and Width** buttons
2. Make sure the [control](#) that is the size you want is selected.
3. Do one of the following:
 - To resize just one control, click the button that you want on the **Alignment and Sizing** toolbar, and then click the control that you want to resize.
 - To resize several controls, lock the button that you want on the **Alignment and Sizing** toolbar by double-clicking it, and then click the controls that you want to resize.

To release the lock, click the same or a different button on the **Alignment and Sizing** toolbar, or press ESC.

[Resize a control to fit its contents](#)

You can resize an a [label](#), [command button](#), [image control](#), or [unbound object frame](#) to fit its contents. Also, if you increase the font size for a control, you can resize the control to accommodate the larger font.

1. Open a form, report, or data access page in [Design view](#).
2. Select the [control](#) or controls that you want to resize.
3. On the **Format** menu, point to **Size**, and then click **To Fit**.

[Resize a control by using the grid](#)

1. Open a form, report, or data access page in [Design view](#).
2. If the grid isn't already displayed, on the **View** menu, click **Grid**.

3. Select the [controls](#) that you want to resize.
4. On the **Format** menu, point to **Size**, and then click **To Grid**.

[Show All](#)

Copy formatting from one control to another

This procedure does not apply to [data access pages](#).

1. Open a form or report in [Design view](#).
2. Click the [control](#) that has the formatting characteristics that you want to copy.

Note You can copy characteristics only from a single control.

3. Click **Format Painter**  on the **Form Design** or **Report Design** toolbar. If you know that you'll want to use the same formatting characteristics on several controls, double-click the button to lock it down.
4. Click the control that you want to paste the formatting characteristics to. You can click the same type of control, or you can select another type of control. For example, you can copy formatting characteristics from a text box to a list box.

Microsoft Access copies any of the following properties:

- **BackColor**
- **BackStyle**
- **BorderColor**
- **BorderStyle**
- **BorderWidth**
- **DisplayWhen**
- **FontItalic**
- **FontName**
- **FontSize**
- **FontUnderline**
- **FontWeight**
- **ForeColor**
- **LabelAlign**
- **LabelX**
- **LabelY**

- **LineSlant**
- **SpecialEffect**
- **TextAlign**
- **Visible**

Note You can't copy properties to or from [ActiveX controls](#).

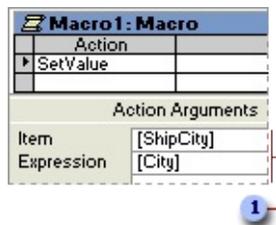
[Show All](#)

Base the value of a control on another control

In a form or report, you can assign the value of a [control](#) based on the value of another control by adding a SetValue action to a [macro](#) or by using an assignment statement in a [procedure](#).

[Set the value of a control based on the value of another control by using a macro](#)

1. Create a macro.
2. In a blank action row, click **SetValue** in the [action list](#).
3. Set the **Item** argument to the [identifier](#) for the control whose value you want to set, and then set the **Expression** argument to the identifier for the control whose value you want to assign to the first control.



❶ These arguments assign the value of the City control to the ShipCity control.

4. Click **Save**  to save the macro.
5. Run the macro.

Note Depending on the context, you might need to use qualified identifiers for the controls. For example, instead of **[ShipCity]**, you might need to use the action argument **Forms![ShipForm]![ShipCity]**.

[Set the value of a control based on the value of another control by using Visual Basic for Applications](#)

1. Create a Visual Basic procedure.

If you want to set the value of control when a particular [event](#) occurs, create an [event procedure](#).

[How?](#)

You can set an [event property](#) for a form, report, or [control](#) to **[Event Procedure]** to run code in response to an event. Microsoft Access creates the [event procedure](#) template for you. You can then add the code you want to run in response to the particular event.

1. Open a form or report in [Design view](#).
2. Display the property sheet for the form or report, or for a section or control on the form or report.
3. Click the **Event** tab.
4. Click the event property for the event that you want to trigger the procedure. For example, to display the event procedure for the **Change** event, click the **OnChange** property.
5. Click **Build** next to the property box to display the **Choose Builder** dialog box.
6. Double-click **Code Builder** to display the event procedure **Sub** and **End Sub** statements in the [form module](#) or [report module](#). These statements define, or declare, the event procedure.

Microsoft Access automatically declares event procedures for each object and event in a form or report module by using the **Private** keyword to indicate that the procedure can be accessed only by other procedures in that module.

7. Add the code to the event procedure that you want to run when the event occurs. For example, to produce a sound through the computer's speaker when data in the CompanyName text box changes, add a **Beep** statement to the CompanyName_Change event procedure, as follows:

```
Private Sub CompanyName_Change()  
    Beep  
End Sub
```

The event procedure runs each time the **Change** event occurs for the object.

2. Add an assignment statement to the procedure by typing the [identifier](#) for the control whose value you want to set, an equal sign (=), and the identifier for the control whose value you want to assign to the first control.

To refer to the value of a control on the current form, type the **Me** keyword followed by the **!** operator and the name of the control. For example, the following statement assigns the value of the City control on the current form to the ShipCity control on the current form:

```
Me![ShipCity] = Me![City]
```

To refer to the value of a control on a different form, type the control's full identifier. For example, the following identifier refers to the ShipCity control on the ShipForm form:

```
Forms![ShipForm]![ShipCity]
```

3. Run the procedure.

[Show All](#)

Create custom colors

If the palette doesn't have the color that you need, you can create your own custom colors.

1. Open a form, report, or data access page in [Design view](#).
2. Select the [section](#) or [control](#), and then click **Properties**  on the **Form Design**, **Report Design** or **Page Design** toolbar to open the property sheet.
3. Do one of the following:
 - In a form or report, click the **ForeColor** (for text in a control), **BackColor**, or **BorderColor** property.
 - In a data access page, click the **Color** (for text in a control), **BackgroundColor**, or **BorderColor** property.
4. Click the **Build** button  next to the property box to open the **Color** dialog box.
5. Click **Define Custom Colors** to expand the dialog box.
6. Click one of the **Custom Color** boxes on the left side of the dialog box.
7. Click on any part of the color refiner box that has the color you want. The color refiner box displays all of the available custom colors.
8. Drag the arrow beside the luminosity bar to define the brightness of the color selected in the color refiner box.

Microsoft Access displays the color you create in the left side of the **Color/Solid** box. If you want to see the solid color most closely resembling the one you've created, double-click the right side of the **Color/Solid** box.

9. Click **Add To Custom Colors**.

Access displays the custom color you created in the **Custom Color** box.

Note You can also create a color by typing numbers in the **Red**, **Green**, **Blue**, **Hue**, **Sat** (saturation), and **Lum** (luminosity) boxes.

[Show All](#)

Turn Snap To Grid behavior on or off

1. Open a form, report, or data access page in [Design view](#).
2. On the **Format** menu, click **Snap To Grid**.

If **Snap To Grid** is turned on, when you create a [control](#) by clicking the form, report, or data access page, Microsoft Access aligns the upper-left corner of the control to the grid. If you create a control by dragging, Access aligns all corners of the control to the grid. If you move or resize an existing control, Access lets you move the control or its boundary from grid point to grid point only.

When **Snap To Grid** is turned off, Access ignores the grid and lets you place, move, or resize controls anywhere on the form, report, or data access page.

Note If you want to temporarily override the current **Snap To Grid** setting, hold down the CTRL key while you're placing, moving, or resizing a control. For example, if **Snap to Grid** is turned on, you can still move a control anywhere on the form, report or data access page by holding down the CTRL key. Releasing the CTRL key restores the current **Snap To Grid** setting.

[Show All](#)

Set a color property to a system color

In a database created using Microsoft Office Access 2003, most views, controls, and dialog boxes automatically inherit the Windows theme, if you are using Microsoft Windows XP SP 1, and the chosen theme is not **Windows Classic**. Exceptions are data access pages, reports, diagrams, SQL view of a query, and modules.

In addition to inheriting the Windows theme, Access also allows you to set the **BackColor**, **BorderColor**, and **ForeColor** properties in a form or report, or the **BackgroundColor**, **BorderColor**, and **Color** properties in a data access page to a system color. Setting color properties to match Windows system colors enables you to specify a single setting that can display different colors on different users' computers, depending on the colors they've chosen in Windows Control Panel.

1. Open a form, report, or data access page in [Design view](#).
2. Open the property sheet for the [section](#) or [control](#).
3. In the property sheet, do one of the following:
 - In a form or report, click the **BackColor**, **BorderColor**, or **ForeColor** property.
 - In a data access page, click the **BackgroundColor**, **BorderColor**, or **Color** property.
4. In the property box, type one of the numbers that are listed in the following table. For example, if you want the background of your form to display the same color that you or another user uses for the Windows background color, set the **BackColor** property to **-2147483643**, the value for Window.

Screen element	Form or report	Data access page
Scroll bar	-2147483648	scrollbar
Desktop	-2147483647	background
Active window title bar	-2147483646	activecaption
Inactive window title bar	-2147483645	inactivecaption
Menu bar	-2147483644	menu
Window	-2147483643	window
Window frame	-2147483642	windowframe
Menu text	-2147483641	menutext

Window text	-2147483640	windowtext
Title bar text	-2147483639	captiontext
Active window border	-2147483638	activeborder
Inactive window border	-2147483637	inactiveborder
Application background	-2147483636	appworkspace
Highlight	-2147483635	highlight
Highlight text	-2147483634	highlighttext
3-D face	-2147483633	buttonface
3-D shadow	-2147483632	buttonshadow
Dimmed (disabled) text	-2147483631	graytext
Button text	-2147483630	buttontext
Inactive window title bar text	-2147483629	inactivecaptiontext
3-D highlight	-2147483628	buttonhighlight
3-D dark shadow	-2147483627	threeddarkshadow
3-D light	-2147483626	threedlightshadow
ToolTip text	-2147483625	infotext
ToolTip background	-2147483624	infobackground

Notes

- The Windows system color value refers only to the color of the screen element listed, not to the type of object it can be assigned to. For example, you could set the **BackColor** property for a text box to the Windows system color for scroll bars, the desktop, or any other screen element.
- The values for the Windows screen elements are always the same, regardless of the user's color settings.

[Show All](#)

Group or ungroup controls

1. Open a form or a report in [Design view](#).

Note You cannot group controls on a data access page.

2. Do one of the following:

[Group controls](#)

1. Select the [controls](#) you want to include in a [group](#). You must select at least two controls to create a group.
2. On the **Format** menu, click **Group**.

[Ungroup controls](#)

1. Click any [control](#) in the [group](#) to select the group.
2. On the **Format** menu, click **Ungroup**.

Notes

- When you ungroup controls, the group rectangle is deleted and all controls are automatically selected.
- You can only ungroup controls when a single group is selected.

[Show All](#)

Select a control that is part of a group

1. Open a form or a report in [Design view](#).
2. Click any [control](#) in the [group](#) to select the group.
3. Click the control you want to select.

Notes

- When you click any control within a group once, the entire group gets selected. The property sheet, if open, displays only those properties common to all the controls within the group.
- When you click a control within a selected group, the control gets selected and the properties for that control are available on the property sheet.

[Show All](#)

Add or remove a control in a group

1. Open a form or a report in [Design view](#).

Note You cannot group controls on a data access page.

2. Do one or more of the following:

[Add a control to an existing group](#)

1. While pressing the SHIFT key, select the [control](#) you want to add to the [group](#), and then click any control in the group to select the group.
2. On the **Format** menu, click **Group**.

[Remove a control from a group](#)

1. Click any [control](#) in the [group](#) to select the group.
2. On the **Format** menu, click **Ungroup**.
3. Click the form background to cancel the selection of controls that were in the group.
4. Select the controls you now want in the group.
5. On the **Format** menu, click **Group**.

[Delete a control that is part of a group](#)

1. Click any [control](#) in the [group](#) to select the group.
2. Click the control you want to delete.
3. Press the DELETE key.

Note If there are only two controls in a group and you delete one control, the group is also deleted.

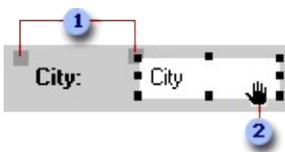
[Show All](#)

Move or resize a control in a group

[Move a control that is part of a group](#)

1. Open a form or a report in [Design view](#).
2. Click any [control](#) in the [group](#) to select the group.
3. Click the control you want to move.
4. Do one of the following:

To move the control and its label, pass the pointer over the border (not the move handle) of the control until it turns into an open hand, and then drag the control and its label to a new position.



- 1 Move handles
- 2 When the pointer changes to an open hand, you can drag the control.

To move the control or its label separately, place the pointer over the move handle in the upper-left corner of the control or label. When the pointer changes to an upward-pointing hand, drag the control or label.



- 1 When the pointer changes to an upward-pointing hand, you can drag the control.

Note If you want to move a control in small increments, you may find it easier to press CTRL plus the appropriate arrow key.

[Resize a control that is part of a group](#)

1. Open a form or a report in [Design view](#).
2. Click any [control](#) in the [group](#) to select the group.
3. Click the control you want to resize.
4. Drag the sizing handles of the control until it is the size you want.



- 1 Move handle
- 2 Sizing handles

Notes

- If you want to make small adjustments to the size of a control, you may find it easier to press SHIFT plus the appropriate arrow key.
- You can resize all the controls in a group by dragging the sizing handles of the group rectangle.

Notes

- When you move or resize a control in a group, the group rectangle is automatically resized to fit around all the controls.
- If **Snap To Grid** is turned on, the group rectangle snaps to the grid when it's moved or resized.

[Show All](#)

About adding a picture or object

You can display pictures in a form, report, or [data access page](#). A picture is just one type of [object](#) that you create in another application and can use in Microsoft Access. In a form or report, you can also display other objects or parts of objects that you've created in another application. For example, you can add a picture that you created with Microsoft Paint, a worksheet that you created with Microsoft Excel, or a document that you created with Microsoft Word. You can add all the information in a file or just selected information.

When you add a picture or object, consider the following:

- Is the picture or object [bound](#) or [unbound](#)? In other words, does it change when you move from record to record?
- Should the picture or object be [embedded](#) or [linked](#)? In other words, is it stored in the [Microsoft Access file](#), or is it stored in an external file and linked to the Access file?

[Bound and unbound pictures and objects](#)

A bound object is a picture or object that is stored in a table. When you move to a new record, the object that is displayed in the form or report changes. For example, you might want to store a picture for each employee in your company.

On the other hand, an unbound object is stored in the design of the form, report, or data access page. When you move to a new record, the object doesn't change. For example, you might want to add a logo that you created with Microsoft Paint to a form or report.

On a data access page, you can bind a picture by entering the path to that picture in a field, and then binding an [image control](#) to the field. Unlike a bound picture on a form or report, a bound picture in a data access page can't be stored in a table—you can store only the path to the picture. On a data access page, you can also insert an unbound picture, such as a logo or banner, which doesn't change when you move to a new record.

[Embedded and linked pictures and objects](#)

In a form or report, a picture or object can be an embedded object or a linked object. Both bound and unbound objects can be either embedded or linked. In a form, you use a [bound object frame](#) to display an embedded or linked bound object, and you use an [unbound object frame](#) to display an embedded or linked unbound object.

When you embed an object in a form or report, Microsoft Access stores the object in your database file— so an embedded object is always available. If you modify the embedded object from your form or report, the object is changed in your database file. If you embed the same picture or object in another form or report, it's added to the database file again.

When you link to an object, you can look at the object and make changes to it from your form or report, but the changes are stored in the original object file, not in your database file. You can also modify the object file separately, in which case the latest changes will appear the next time you open the form or report.

Linking to an object is useful for very large files that you don't want to include in your database file, and for files that you want to use in several forms or reports. If a linked object file is moved, you have to reconnect the link.

You can specify whether the latest changes appear in a linked object automatically, or only when you request them. If you are using an unbound object frame, those changes appear when you open the frame; if you are using a bound object frame, those changes appear when the control gets the [focus](#).

On a data access page, you can't embed or link objects, because you can't use unbound object frames or bound object frames— you can add pictures only by using the [image control](#). A picture in an image control has no connection with the source application used to create it.

[Examples of bound or unbound and embedded or linked objects](#)

**Example
object**

**Type of
control
to use**

**Embedded
or linked?**

Embedded (if disk space

A logo on an invoice report that you don't need to update	Image control	isn't an issue; otherwise linked)
A picture on a form or report that you expect to update	Unbound object frame	Embedded
Part of a Microsoft Excel worksheet in a sales report	Unbound object frame	Linked
A picture for each employee to be stored in an Employees table	Bound object frame	Embedded
Employee résumés (created and stored in a Word document) available from an Employees form	Bound object frame	Linked, and displayed as an icon

[Further considerations when adding pictures](#)

An unbound picture on a form, report, or data access page can appear as a [watermark](#), a background picture that covers the entire window. An unbound picture can also appear in a [control](#).

If you want to add an unbound picture that appears in a control on a form or report, you use an [image control](#) or an [unbound object frame](#). Whether you use an image control or an unbound object frame depends on whether you want to be able to edit the picture directly from the form or report. If you won't need to update the picture, you can add the picture by using an image control. Use an unbound object frame if you will need to update the picture. For most unbound pictures, you'll probably want to use an image control, because image controls load much faster than unbound object frames.

When you use a bound control to display a picture, you see a different picture for each record. The best way to display a different picture for each record is to store the paths to the pictures in a Text field. You can also store pictures in an [OLE Object field](#) in a [Microsoft Access database](#), or in an [image column](#) in a SQL Server database to which an [Access project](#) is connected.

Security Note Users of your database or data access page will be able to view the image paths. Make sure that the paths do not include personal information,

such as your name or the name of your computer.

If you want to display a different picture for each record in a data access page, you must store the locations of the pictures in a Text field— a data access page can't display the contents of an OLE Object field or an image column. Another advantage of storing paths to pictures instead of storing the pictures themselves is that the size of your database doesn't increase as rapidly. If you use a Text field to store the paths to the pictures, you must use Visual Basic [event procedures](#) to display the pictures in a form or report.

You can use only an image control to add a picture to a data access page. After you add a picture to a data access page, you won't be able to edit it. When you insert a picture in a data access page, the picture is added to the supporting files that accompany the data access page; the picture is not stored in the data access page. If you want to move a data access page to a new location, do not copy the data access page. Instead, save the data access page to the new location; the picture is automatically saved with it.

If you want a data access page to use a picture in a location that you specify instead of a picture that has been added to supporting files, you can link the picture. Linking a picture is a good idea if you are going to use a picture in more than one data access page.



[Bitmaps and metafiles, and other graphics files that you can use to display a picture](#)

You can use [bitmap](#) files (.bmp or .dib extensions), [metafiles](#) (.wmf or .emf extensions), or other graphic files such as [GIF](#) and [JPEG](#) to display background pictures or pictures in [bound object frames](#), [unbound object frames](#), or [image controls](#).

When to use bitmaps

- If scaling and disk space aren't concerns, or if you don't have access to a tool that creates metafiles, use a bitmap.
- If you want to create a tiled background picture, a bitmap will work well because it won't be scaled.
- If you want to be able to make small changes to the picture, use a bitmap. With bitmaps you have [pixel-level](#) control, so that you can make very small adjustments to the picture.

When to use metafiles

- If you need to be able to adjust the size of your picture, it's better to use a metafile. Metafiles scale better than bitmaps.
- Metafiles are usually much smaller than bitmaps, so if you need to conserve disk space, use a metafile.

Many graphics programs allow you to save files as metafiles. See the documentation that came with your graphics program for details. Note that if you paste a bitmap into a graphics program and save it as a metafile, it will still be a bitmap— it's just embedded in the metafile.

When to use GIF and JPEG files

- If you need to be able to view a picture in a data access page that is available on an [intranet](#) or the [Internet](#), use a GIF or JPEG file.

[Show All](#)

Add a picture or object

You can add two types of pictures or objects: an [unbound](#) picture or object, which doesn't change from record to record, or a [bound](#) picture or object, which can be different as you move from record to record. A picture or object can also be [embedded](#) or [linked](#).

Add a picture or object that doesn't change from record to record

In a form, report, or [data access page](#), you can add a background picture that simulates a [watermark](#) or a picture that is displayed in an [image control](#). You can also use an [unbound object frame](#) to add a picture or object to a form or report.

[Add a background picture](#)

1. Open the form, report, or data access page in [Design view](#).
2. Do one of the following:

[Add a background picture to a form or report](#)

1. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
2. In the **Picture** property box on the **Format** tab, specify the path and file name for a .bmp, .ico, .dib, .wmf, .emf, or other image file. If you've installed graphics filters for other applications, you can use any file supported by those filters. If you're not sure of the path or file name, click the **Build** button  to use the **Insert Picture** dialog box to locate your image.
3. In the **PictureType** property box, specify whether you want the picture to be embedded or linked.
4. After you add the picture to the form or report, you can size or clip it by clicking one of the following settings in the **PictureSizeMode** property box.

Setting

Description

Clip Displays the picture at actual size. If the picture is larger than the Form window, or if it is larger than the area within the

margins of the report, the image is cut off.

Stretch Sizes the picture to fit the Form window or to fit within the margins of the report. This setting may distort the image.

Zoom Sizes the picture to fill either the height or width of the Form window, or the height or width of the area within the margins of the report. This setting won't clip the picture or distort its proportions.

5. If you want to change the alignment of the picture, click one of the settings in the **PictureAlignment** property box.

Note If you want to center the background picture on a form, and you want the background picture to resize when you size the window, click the **Center** setting. If you don't want the background picture to resize when you size the window, click the **Form Center** setting.

6. If you want to repeat the picture across the background of the form, set the **PictureTiling** property to **Yes**. Tiling starts at the position that is specified for the **PictureAlignment** property.

Note Set the **PictureSizeMode** property to **Clip** if you want to tile the background picture. The background picture will not be tiled if the **PictureSizeMode** property is set to **Zoom** or **Stretch**.

7. If necessary, specify which pages of the report the picture will appear on by setting the **PicturePages** property. The available settings are **All Pages**, **First Page**, and **No Pages**.

 [Add a background picture to a data access page](#)

1. On the **Format** menu, point to **Background**, and then click **Picture**.
2. In the **Insert Picture** dialog box, make sure that **Picture Files** is selected in the **Files of type** list, and then locate the file that you want to use for the background.

Note You can set the following properties in the property sheet to adjust how the background picture is displayed: **BackgroundPositionX**, **BackgroundPositionY**, and **BackgroundPositionRepeat**.

 [Use an image control to add an unbound picture](#)

1. Open the form, report, or data access page in [Design view](#).
2. Click the **Image** tool  in the [toolbox](#).
3. On the form, report, or data access page, click where you want to place the picture.
4. In the **Insert Picture** dialog box, specify in the **File name** box the path and file name for picture. You might need to use the **Look In** box to specify the drive and/or folder where the picture is located.

If you want to link to the picture in a data access page that you will use over an [intranet](#) or the [Internet](#), specify a [URL](#) that is a valid Web address in the **File name** box.

5. Do one of the following:
 - To add the picture to a form or report, click **OK**.
 - To add the picture to a data access page, click **Insert**.
 - To link the picture in a data access page, click the arrow next to **Insert**, and then click **Link to File**.

Microsoft Access creates an image control and displays the picture in the control.

6. To link the picture in a form or report, make sure the control is still selected, and click **Properties** . Then set the **PictureType** property for the image control to **Linked**.

Notes

- When you add a picture to an image control in a form or report, Microsoft Access embeds the picture by default. If you plan to use the same picture on several forms or reports, you may want to link the picture so that it isn't stored separately in each form or report.
- When you add a picture in a data access page, the picture is added to the supporting files that Access saves to the same location as the page. If you plan to use the same picture on several different data access pages, and you don't want the picture to be added to the supporting files several times, you want to link to the image file.

 [Create and embed an unbound object on a form or report](#)

1. Open a form or report in [Design view](#).
2. Click the **Unbound Object Frame** tool  in the [toolbox](#).
3. On the form or report, click where you want to insert the object.
4. In the **Insert Object** dialog box, click **Create New** if it's not already selected, and then in the **Object Type** box, click the object type you want to create.
5. Select the **Display as Icon** check box if you want the object to appear as the icon for the application used to create the object instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
6. Click **OK**. Microsoft Access creates an [unbound object frame](#).
7. Create the object by using the application that you chose in the **Object Type** box in step 4, and then do one of the following:
 - If you created the object [in place](#), click the form outside the [bound object frame](#).

Note If you click **Exit** on the **File** menu, you'll quit Microsoft Access.

- If the object that you created opened in a separate window, click **Exit** on the **File** menu of that separate window. If you're prompted to update the document, click **OK**.

8. To enable editing of the object in [Form view](#), set the **Enabled** property of the unbound object frame to **Yes** and the **Locked** property to **No**. These properties are found on the **Data** tab of the form's property sheet, which can be accessed by double-clicking the [form selector](#).

[Embed an existing unbound object or part of an existing unbound object on a form or report](#)

In a form or report, you can embed an existing object that was created in another application. You can also embed part of an existing object if the application that you're copying from has [OLE](#) support for copying part of an existing document.

1. Open a form or report in [Design view](#).
2. Do one of the following:

[Embed an existing unbound object](#)

If the application that you're copying from supports OLE drag-and-drop editing, you can drag the file directly from My Computer or the desktop. Otherwise, do the following:

1. Click the **Unbound Object Frame** tool  in the [toolbox](#).
2. On the form or report, click where you want to insert the object.
3. In the **Insert Object** dialog box, click **Create from File**, and then specify a path to the file. If you don't know the path, click **Browse**.
4. Select the **Display as Icon** check box if you want the object to appear as the icon for the application where the object was created instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
5. Click **OK**.

Microsoft Access creates an unbound object frame and displays the object in the frame.

[Embed part of an existing unbound object](#)

If the application that you're copying from supports OLE drag-and-drop editing, you can drag selected information from that application to the Design view of a Microsoft Access form or report. When you do this, Access creates an unbound object frame and inserts the selected information in the frame. Otherwise, do the following:

1. Open the file in the application that you used to create it, and select the information you want to insert. For example, open Microsoft Excel and select a range of cells.
2. On the **Edit** menu, click **Copy**.
3. Switch to Microsoft Access.
4. In [Design view](#) of the form or report, do one of the following:
 - To embed the information, click **Paste**  on the toolbar.
 - To display the object as an icon or specify what type of object to paste as, click **Paste Special** on the **Edit** menu.

Access creates an unbound object frame, and displays the object in the frame.

3. To enable editing of the object in [Form view](#), select the control, click **Properties** on the toolbar, and on the **Data** tab, set the **Enabled** property to **Yes** and the **Locked** property to **No**.

[Create a link to an existing unbound object or part of an existing unbound object on a form or report](#)

In a form or report, you can create a [link](#) to an existing object that was created in another application. You can also create a link to part of an existing object if the application that you're linking to has [OLE](#) support to do so.

1. Open a form or report in [Design view](#).
2. Do one of the following:

[Create a link to an existing unbound object](#)

1. Click the **Unbound Object Frame** tool  in the [toolbox](#).
2. On the form or report, click where you want to place the object.
3. In the **Insert Object** dialog box, click **Create from File**, and then specify a path to the file. If you don't know the path, click **Browse**.
4. Select the **Link** check box.
5. Select the **Display as Icon** check box if you want the object to appear as the icon for the application where the object was created instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
6. Click **OK**.

Microsoft Access creates an unbound object frame and displays the object in the frame.

[Create a link to part of an existing unbound object](#)

1. Open the file in the application that you used to create it, and select the information you want to link to. For example, open Microsoft Excel and select a range of cells.
2. On the **Edit** menu, click **Copy**.
3. Leave the application open, and switch to Microsoft Access.

4. In [Design view](#) of the form or report, click **Paste Special** on the **Edit** menu.
5. In the **Paste Special** dialog box, click **Paste Link**, and then click the type of object to paste as in the **As** list.
6. Select the **Display as Icon** check box if you want the object to appear as the icon for the application where the object was created instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
7. Click **OK**.

Microsoft Access creates an unbound object frame and displays the object in the frame.

3. To enable editing of the object in [Form view](#), select the control, click **Properties** on the toolbar, and on the **Data** tab, set the **Enabled** property to **Yes** and the **Locked** property to **No**.

Note When you link an object, the link is automatic by default. This means that if the object is updated outside Microsoft Access, the changes will automatically appear the next time you open the form or report. If you want to control when the data is refreshed, you can change the link from automatic to manual.

Add a picture or object that changes from record to record

You can use an [image control](#) to display a bound picture in a form, report, or data access page. To display a bound embedded or linked picture or object on a form or report, you must create a [bound object frame](#).

 [Add a bound image control](#)

1. Do one of the following:
 - To display pictures that change from record to record without storing them in a table, create a text field to store the locations of the image files. You must store the locations of the pictures in a text field if you want to display the pictures in a data access page. You can also use a text field to store the location of pictures and then use those pictures in a form or report, but you must use Visual Basic [event procedures](#) to

display the pictures.

- To display pictures that are stored in a table, do one of the following:
 - To store pictures in a [Microsoft Access database](#), create an [OLE Object field](#).
 - In a [Microsoft Access project](#), create an [image column](#) to store the pictures.

You can't display the contents of an OLE Object field or an image column in a data access page.

2. Open the form, report, or data access page in [Design view](#).
3. Do one of the following:

[Add an image control to a data access page, or, in a form or report, add an image control that is bound to an OLE Object field or image column](#)

1. In the toolbox, click the **Image** tool .
2. If the field list isn't displayed, click **Field List**  on the toolbar.
3. From the field list, drag the field that contains the pictures or the locations of the pictures to the form, report, or data access page. Place the upper-left corner of the icon where you want the upper-left corner of the image to be, and release the mouse button.

[In a form or report, add an image control that is bound to a text field containing the paths to the pictures](#)

1. If the field list isn't displayed, click **Field List**  on the toolbar.
2. From the field list, drag the field that contains the locations of the pictures to the form or report.
3. In the toolbox, click the **Image** tool .
4. On the form or report, click where you want to place the object.
5. In the **Insert Picture** dialog box, specify the path to any picture, and then click **OK**. For example, you can type the path to the picture for the first record.
6. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
7. Click the **Build** button  next to the **OnCurrent** property box, and then click **Code Builder** in the **Choose Builder** dialog box.

8. Create the following event procedure. Substitute the name of the image control on your form or report for *ImageControlName* and the name of the control containing the path for *ImagePath*.

```
Private Sub Form_Current()  
On Error Resume Next  
Me![ImageControlName].Picture = Me![ImagePa  
End Sub
```

Note that if you are adding the event procedure in a report, the first line of the procedure will be the following:

```
Private Sub Report_Current()
```

9. On the **File** menu in the Visual Basic Editor, click **Close and return to Microsoft Access**.
10. Click the control that is bound to the text field containing the location of the pictures, and then click **Properties**  on the toolbar to open the property sheet.
11. Click the **Build** button  next to the **AfterUpdate** property box, click **Code Builder** in the **Choose Builder** dialog box, and create the following event procedure. Substitute the name of the image control on your form or report for *ImageControlName* and the name of the control containing the path for *ImagePath*.

```
Private Sub ImagePath_AfterUpdate()  
On Error Resume Next  
Me![ImageControlName].Picture = Me![ImagePa  
End Sub
```

This event procedure enables you to add or change a picture location in Form view.

12. On the **File** menu in the Visual Basic Editor, click **Close and return to Microsoft Access**.

Note To see an example of an image control that is bound to a field containing the locations of pictures, open the Employees form in the Northwind sample database.

[Create a bound object frame](#)

1. If you haven't already done so, create a field in a table, and set the field's data type to [OLE Object](#) in a [Microsoft Access database](#), or to [image](#) in a [Microsoft Access project](#).
2. Create a form or report (or open an existing form or report) that includes the OLE Object field or image column in its record source.
3. In form or report [Design view](#), click **Field List**  on the toolbar to display the field list.
4. Drag the OLE Object field or image column from the field list to the form or report.

Microsoft Access creates a bound object frame that's bound to the OLE Object field or image column.

Note If you've already created the control by clicking the **Bound Object Frame**  tool in the toolbox, you can bind the control to a field in an underlying record source by setting the **ControlSource** property for the bound object frame to the name of the field.

[Create and embed a bound object on a form or datasheet](#)

1. Do one of the following:
 - Open a form in form [Design view](#), create a bound object frame, and then switch to [Form view](#) or [Datasheet view](#).
 - Open a table or query in Datasheet view.
2. Move to the record in which you want to insert the object, and click the [OLE Object field](#) in a [Microsoft Access database](#), or the [image column](#) in a [Microsoft Access project](#).
3. On the **Insert** menu, click **Object**.
4. In the **Insert Object** dialog box, click **Create New** if it's not already selected, and then click the kind of object that you want to create in the **Object Type** box.
5. Select the **Display as Icon** check box if you want the object to appear as an icon instead of the object itself. Displaying an object as an icon can be

helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.

6. Click **OK**.
7. Create the object by using the application that you chose in the **Object Type** box in step 4, and then do one of the following:
 - If you created the object in place, click the form outside the [bound object frame](#).

Note If you click **Exit** on the **File** menu, you'll quit Microsoft Access.

- If the object that you created opened in a separate window, click **Exit** on the **File** menu of that separate window. If you're prompted to update the document, click **OK**.

If you're in Form view, Access displays the object. If you're in Datasheet view, Access displays the name of the object, for example, "Microsoft Excel Worksheet" or "Bitmap Image." To see the object that you have inserted, you must create a bound object frame on a form or report.

If you are in a form and you need to edit the object, double-click it.

 [Embed an existing bound object or part of an existing bound object on a form or datasheet](#)

In a form or datasheet, you can embed an existing object that was created in another application. You can also embed part of an existing object if the application that you're copying from has [OLE](#) support for copying part of an existing document.

1. Do one of the following:
 - Open a form in form [Design view](#), create a bound object frame, and then switch to [Form view](#) or [Datasheet view](#).
 - Open a table or query in Datasheet view.
2. Do one of the following:

 [Embed an existing bound object](#)

If the application that you're copying from supports OLE drag-and-drop

editing, then, instead of using the **Object** command, you can drag the file directly from My Computer or the desktop to the [OLE Object field](#) in a [Microsoft Access database](#), or to the [image column](#) in a [Microsoft Access project](#). Otherwise, do the following:

1. Move to the record in which you want to insert the object and click the OLE Object field in a Microsoft Access database (.mdb) or the image column in a Microsoft Access project (.adp).
2. On the **Insert** menu, click **Object**.
3. In the **Insert Object** dialog box, click **Create from file**, and then specify a path to the file. If you don't know the path, click **Browse**.
4. Select the **Display as Icon** check box if you want the object to appear as an icon instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
5. Click **OK**.

[Embed part of an existing bound object](#)

If the application that you're copying from supports OLE drag-and-drop editing, you can drag selected information from that application to the [OLE Object field](#) in a [Microsoft Access database](#), or to the [image column](#) in a [Microsoft Access project](#). Otherwise, do the following:

1. Switch to the application that you used to create the object, and select the information that you want to insert. For example, open Microsoft Excel and select a range of cells.
2. On the **Edit** menu, click **Copy**.
3. Switch back to Access, move to the record that you want to paste the object into, and click the OLE Object field in a Microsoft Access database (.mdb) or the image column in a Microsoft Access project (.adp).
4. Click **Paste**  on the toolbar.

Note If you want to display the object as an icon, or if you want to specify what type of object to paste as, click **Paste Special** on the **Edit** menu.

If you're in Form view, Microsoft Access displays the object. If you're in Datasheet view, Access displays the name of the object, for example "Microsoft Excel Worksheet" or "Bitmap Image." To see the object, you must create a bound object frame on a form or report.

[Create a link to an existing bound object or to part of an existing object on a form or datasheet](#)

In a form or datasheet, you can create a link to an existing object that was created in another application. You can also link to part of an existing object if the application that you're linking to has [OLE](#) support to do so.

1. Do one of the following:
 - Open a form in form [Design view](#), create a bound object frame, and then switch to [Form view](#) or [Datasheet view](#).
 - Open a table or query in Datasheet view.
2. Do one of the following:

[Create a link to an existing bound object](#)

1. Move to the record in which you want to link the object, and click the [OLE Object field](#) in a [Microsoft Access database](#), or the [image column](#) in a [Microsoft Access project](#).
2. On the **Insert** menu, click **Object**.
3. In the **Insert Object** dialog box, click **Create from File**, and then specify a path to the file. If you don't know the path, click **Browse**.
4. Select the **Link** check box.
5. Select the **Display as Icon** check box if you want the object to appear as the icon for the application where the object was created instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
6. Click **OK**.

[Create a link to part of an existing bound object](#)

You can use this procedure if the application that you're linking to has OLE support for linking to part of an existing document.

1. Switch to the application that you used to create the object and select the information you want to link. For example, open Microsoft Excel and select a range of cells.
2. On the **Edit** menu, click **Copy**.
3. Leave the application open, and switch back to Microsoft Access.
4. Move to the record you want to paste the object into, and click the [OLE Object field](#) in a [Microsoft Access database](#), or the [image column](#) in a [Microsoft Access project](#).
5. On the **Edit** menu, click **Paste Special**.
6. In the **Paste Special** dialog box, click **Paste Link**, and then click the type of object to paste as in the **As** list.
7. Select the **Display as Icon** check box if you want the object to appear as the icon for the application where the object was created instead of as the object itself. Displaying an object as an icon can be helpful when an object contains supplementary information that doesn't have to be displayed. Displaying an object as an icon also uses significantly less disk space.
8. Click **OK**.

Notes

- If you're in Form view, Microsoft Access displays the object. If you're in Datasheet view, Microsoft Access displays the name of the object, for example, "Microsoft Excel Worksheet" or "Bitmap Image." To see the object, you must create a bound object frame on a form or report.
- When you link an object, the link is automatic by default. Microsoft Access automatically updates the object when you move to the field. If you want to control when the link is updated, you can change the link from automatic to manual.

Security Note Users of your database or data access page will be able to view the image paths. Make sure that the paths do not include personal information, such as your name or the name of your computer.

[Show All](#)

Delete a picture or object

[Delete a background picture](#)

1. Open the form, report, or data access page in [Design view](#).
2. Do one of the following:

Delete a background picture from a form or report

1. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
2. Clear the **Picture** property box.

Delete a background picture from a data access page

1. On the **Edit** menu, click **Select Page**, and then click **Properties** .
2. Clear the **BackgroundImage** property box.

[Delete a picture or object in a control](#)

1. Do one of the following:
 - If you want to delete an [unbound](#) picture or object, open the form, report, or data access page in [Design view](#) and click the [image control](#) or [unbound object frame](#).
 - If you want to delete a [bound](#) picture or object, open the form in [Form view](#) (or open a datasheet), find the record you want to change, and click the object.
2. Press the DELETE key.

If the picture or object is bound, then Microsoft Access removes it from the [bound object frame](#) and the underlying record source, but the bound object frame remains on the form or report. To delete the frame, open the form or report in Design view, click the control, and press the DELETE key.

3. If you have deleted a bound object frame, and you also want to delete all the objects stored in the underlying field, do one of the following:

In a [Microsoft Access database](#), delete the [OLE Object field](#) in table Design view.

In a [Microsoft Access project](#), delete the [image](#) column in Column Properties view in the [database diagram](#).

[Show All](#)

Edit a picture or object

To edit an object in [Form view](#), set the **Enabled** property of the [unbound object frame](#) or [bound object frame](#) to **Yes**, and set the **Locked** property to **No**. These procedures do not apply to [image controls](#).



[Edit an unbound picture or object in a form or report](#)

1. Open the form or report in [Design view](#).
2. Double-click the object.
3. Microsoft Access opens the application that was used to create the object (the [OLE server](#) application) if it's available.

Note When you double-click a sound or video object, Microsoft Access plays the object. To open these types of objects for editing, click the object, and then click the appropriate **Object** command on the **Edit** menu. For example, point to the **Media Clip Object** command, and then click the **Open** or **Edit** command on the submenu.

4. Make the changes you want and then do one of the following:
 - If you're editing the object in place, click the form outside the unbound object frame. If you click **Exit** on the **File** menu, you'll quit Microsoft Access.
 - If you're editing the object in a separate window, click **Exit** on the **File** menu of the OLE server application. If you're prompted to update the document, click **OK**.



[Edit a bound picture or object on a form or in a datasheet](#)

1. Open the form in Form view, or open a [datasheet](#).
2. Go to the record that contains the object you want to edit.
3. Double-click the object.

Microsoft Access opens the application that was used to create the object (the [OLE server](#) application) if it's available. The application may open in a separate window or you may be able to edit the object [in place](#).

Note When you double-click a sound or video object, Microsoft Access plays the object. To open these types of objects for editing, click the object and then click the appropriate **Object** command on the **Edit** menu. For example, point to the **Media Clip Object** command, and then click the **Open** or **Edit** command on the submenu.

4. Make the changes that you want, and then do one of the following:
 - If you're editing in place on a form, click the form outside the bound object frame. If you click **Exit** on the **File** menu, you'll quit Microsoft Access.
 - If you're editing the object in a separate window, click **Exit** on the **File** menu of the OLE server application. If you're prompted to update the document, click **OK**.

[Control how a picture or object is activated for editing](#)

1. Open the form in [Design view](#).
2. Select the unbound object frame or bound object frame, and then click **Properties** .
3. Do one or more of the following:

To specify whether you activate the object by double-clicking it or by selecting it, set the **AutoActivate** property. For more information, click the **AutoActivate** property box and press F1.

To specify whether you edit the object [in place](#) or in a separate window, set the **Verb** property. For more information, click the **Verb** property box and press F1.

[Show All](#)

Make a picture or object read-only

If a picture or object exists in a [bound](#) or [unbound object frame](#) on a form or report, you can prevent users from editing the picture or object in the [OLE server](#) application. To improve performance, you can also make the picture or object read-only by converting it to an image.

[Prevent users from editing a bound or unbound object in the OLE server application](#)

1. Open the form or report in [Design view](#).
2. Click the object, and then click **Properties**  on the **Form Design** or **Report Design** toolbar.
3. Set the **Locked** property to **Yes** and the **Enabled** property to **No**.

[Convert an unbound picture or object to a read-only image](#)

1. Open the form or report in [Design view](#).
2. Click the object.
3. On the **Format** menu, point to **Change To**, and then click **Image**.

Microsoft Access converts the unbound object frame to an [image control](#). The object remains on the form or report, but you can't open the application that was used to create the object (the OLE server application) from the form or report. If the object is a sound object, just the icon remains. If it's a video object, just the first frame of the video remains. You can't play the sound or video object.

[Convert a bound picture or object to a read-only image](#)

1. Open the form in [Form view](#).
2. Click the object and then click **Copy**  on the **Form Design** toolbar.
3. On the **Edit** menu, click **Paste Special**.
4. In the **As** list, click **Picture**, or one of the bitmap choices, depending on whether you want to create a [metafile](#) or a bitmap.
5. Microsoft Access converts the object to an image. The bound object frame

remains on the form, but you can't open the application that was used to create the object (the OLE server application) from the form. If the object is a sound object, just the icon remains. If it's a video object, just the first frame of the video remains. You can't play the sound or video object.

Note When you make any changes to an object in a bound object frame, you're changing the object only for the current record. To make a change in a different record, move to that record and repeat the process.

[Show All](#)

Show an icon in place of a picture or object

Displaying an object as an application icon can save a significant amount of disk space because the [OLE server](#) (the application that was used to create the object) sends the object with rendering information that consists of only the icon, rather than the complete object. This procedure does not apply to [data access pages](#).

1. Do one of the following:
 - If the object is [unbound](#), click the object in form or report [Design view](#).
 - If the object is [bound](#), open the form in [Form view](#) (or open a datasheet), find the record that you want to change, and click the object.
2. On the **Edit** menu, point to the appropriate **Object** command (for example, point to **Worksheet Object** for a Microsoft Excel object), and then click **Convert**.
3. In the **Convert** dialog box, select the **Display As Icon** check box.

[Show All](#)

Resize a picture

How you adjust the size and proportions of a picture depends on whether it's a background picture (watermark) or a picture that uses an [image control](#), [unbound object frame](#), or [bound object frame](#).

1. Open a form, report, or data access page in [Design view](#).
2. Do one of the following:

[Adjust the size of a background picture on a form or report](#)

1. Double-click the [form selector](#) or [report selector](#) to open the property sheet.
2. Click the **Format** tab, and in the **PictureSizeMode** property box, click one of the following settings:
 - **Clip** Displays the picture at actual size. If the picture is larger than the form or report, the image is cut off.
 - **Stretch** Sizes the picture to fit within the form or report. This setting may distort the image.
 - **Zoom** Displays the picture in its entirety after sizing it to fill either the height or width of the form or report. This setting won't distort the image.

Note To fill the background with multiple copies of the picture, set the **PictureTiling** property to **Yes** and the **PictureSizeMode** property to **Clip**.

[Adjust the size of a picture displayed in an image control, unbound object frame, or bound object frame on a form or report](#)

1. Make sure that the image control, unbound object frame, or bound object frame is selected.
2. Click **Properties**  on the toolbar to display the property sheet.
3. Click the **Format** tab, and in the **SizeMode** property box, click one of the following settings:
 - **Clip** Displays the picture at actual size. If the picture is larger

than the area within the control, the image is cut off.

- **Stretch** Sizes the picture to fit the area within the control. This setting may distort the image.
- **Zoom** Displays the picture in its entirety after sizing it to fill either the height or width of the area within the control. This setting won't distort the image.

 [Adjust the size of a picture displayed in an image control on a data access page](#)

1. Select the [image control](#).
2. Click **Properties**  on the toolbar to display the property sheet.
3. In the **Height** and **Width** property boxes, type the number of [pixels](#) (px), centimeters (cm), inches (in), or [points](#) (pt) that you want for the height and width of the image that is displayed in the image control.

 [Tip](#)

To resize the image control so that its height and width match the dimensions of the picture, point to **Size** on the **Format** menu and click **To Fit**.

[Show All](#)

Show alternate text for a picture

You may want to add alternate text to a picture on a data access page; if the user has graphics turned off in the Web browser, or the user has a screen reader, the alternate text will be displayed in place of the picture.

1. Open a data access page in [Design view](#).
2. Select the [image control](#), and then click **Properties**  on the **Page Design** toolbar to display the control's property sheet.
3. In the **Alt** property box, type the text that you want to be displayed.

[Show All](#)

Update a link to a picture or object

You may lose a link to a picture or object on a form or report if you rename or move the linked picture or object. How you update the link depends on whether it's an object in a [bound object frame](#) or [unbound object frame](#), a picture in an [image control](#), or a background picture (watermark).

[Update a link to a picture or object in a bound or unbound object frame](#)

1. Do one of the following:
 - If the object is [unbound](#), click the object in [form](#) or [report Design view](#).
 - If the object is [bound](#), open the form in [Form view](#) (or open a datasheet), find the record you want to change, and click the object.
2. On the **Edit** menu, click **OLE/DDE Links**.
3. In the **Links** box, click the link you want to reconnect or change.
4. Click **Change Source**.
5. In the **Change Source** dialog box, select the new file name. You may need to switch to a different drive and/or folder to find the appropriate file.
6. Click **Open** to complete the link and close the **Change Source** dialog box, and then click **Close** in the **Links** dialog box.

Note When you make any changes to a link for a [bound object frame](#), you are changing the link only for the current record. To make a change in a different record, move to that record and repeat the process.

[Update a link to a background picture or a picture in an image control](#)

1. Open the form or report in [Design view](#).
2. Do one of the following:
 - If you are updating the link to a background picture, double-click the [form selector](#) or the [report selector](#) to open the property sheet for the form or report.
 - If you are updating the link to a picture in an image control, make sure that the control is selected, and then click **Properties**  on the **Report Design** toolbar.

3. Specify the correct path for the picture in the **Picture** property box.

[Show All](#)

Add a picture to a command or toggle button

 [Add a picture to a command button or a toggle button on a form](#)

1. Open a form in [Design view](#).
2. Click the command button or toggle button, and then click **Properties**  on the toolbar to open its property sheet.
3. Click the **Format** tab, and in the **Picture** property box, type the path and file name for a .bmp, .ico, or .dib file. If you're not sure of the path or file name, click the **Build** button  to open the Picture Builder.

 [Tip](#)

In Microsoft Paint, you can create a file with a picture and caption for a button. Draw the picture and caption on a background of any size, and then select only the portion that will fit on the button. On the **Edit** menu, click **Copy To** and specify a file name in the **Copy To** dialog box.

 [Add a picture to a command button on a data access page](#)

1. Open a data access page in [Design view](#).
2. Click the command button you want to add the picture to, and then click **Properties**  on the toolbar.
3. Click the **Format** tab, and in the **BackgroundImage** property box, specify the location of image you want to use in the following format:

`url("http://WebServerName/FileName")`

4. Do one of the following:
 - To specify that only one image appears on the button, set the **BackgroundRepeat** property to **no-repeat**.

- To specify that copies of the image appear in a horizontal line on the button, set the **BackgroundRepeat** property to **repeat-x**.
 - To specify that copies of the image appear in a vertical line on the button, set the **BackgroundRepeat** property to **repeat-y**.
5. Set the position of the image in the **BackgroundPositionX** and **BackgroundPostionY** property boxes.

[Show All](#)

Convert an OLE object to a different version

1. Do one of the following:
 - If the object is [unbound](#), open the form or report in [Design view](#) and click the object.
 - If the object is [bound](#), open the form in [Form view](#) (or open a datasheet), find the record that you want to change, and click the object.
2. On the **Edit** menu, point to the appropriate **Object** command (for example, point to **Worksheet Object** for a Microsoft Excel object), and then click **Convert**.
3. In the **Convert** dialog box, click the version that you want to convert to. For example, you could convert a Microsoft Graph 97 Chart to a Graph 2000 chart.

[Show All](#)

About charts

Charts are visually appealing and make it easy for users to see comparisons, patterns, and trends in data. In Microsoft Access, you can create charts in a form, report, or data access page. You can also display a datasheet or form in [PivotChart view](#).



[Charts in a form or report](#)

In a form or report, you can use Microsoft Graph to create a chart linked to data from the available tables, [queries](#), or other [record sources](#). The charts can be global (including all data) or record-bound (reflecting data only from the current record, in which case the chart updates as you move from record to record).

Several chart types, including two-dimensional (2-D) and three-dimensional (3-D) types, are available for you to choose from. Formatting options include adding text and graphics to the charts, changing colors, and adding trendlines and error bars.

[Charts in a data access page](#)

You can use Microsoft Office Chart (a Microsoft Office Web Component) to create dynamic, interactive charts in a data access page that you can then make available on the Web. For example, you can create a chart by using data in a table that stores sales figures. When you update the sales figures, the chart updates— so users can always see the latest information on their Web pages.

You can create a chart in a page by using data from tables or queries, or from a [Spreadsheet Component](#) or [PivotTable list](#). You can choose from a variety of chart types in which you can add text, change colors, and take advantage of other formatting options, such as adding trendlines and error bars.

You can also modify a chart in a page that was published as an interactive chart from Microsoft Excel. When you publish an interactive chart in Excel, the data for the chart is published with it as a Spreadsheet Component or PivotTable list. For more information about publishing charts from Excel, see Microsoft Excel Help.

Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later is required to create a chart on a page.

 [PivotChart view of a datasheet or form](#)

In an [Access database](#) or [Access project](#), you can display a datasheet or form that contains sales figures as a dynamic, interactive chart in PivotChart view. PivotChart view automatically uses Microsoft Office Chart (a Microsoft Office Web Component) to display the chart. In PivotChart view, you can choose from a variety of chart types in which you can add text, change colors, and take advantage of other formatting options, such as adding trendlines and error bars.

[Show All](#)

Add a chart to a form or report

When you use the Chart Wizard to create a chart, it uses data you specify to determine whether it should create a global chart that shows all fields, or a record-bound chart that shows only the data in the current record as you move from record to record. If the Chart Wizard creates a global chart, and you want a record-bound chart, you can link the chart to a specific record. To use the Chart Wizard, you must have Microsoft Graph.

Note In a [Microsoft Access project](#), you can't use the Chart Wizard, but you can add a chart by inserting the **Microsoft Graph Chart** object (**Insert** menu, **Object** command).

 [Add a new chart](#)

1. Open the form or report in [Design view](#).
2. On the **Insert** menu, click **Chart**.

If you create a lot of charts, you may want to add the **Chart**  button to the toolbar.

 [How?](#)

Note You can add the **Chart** button to a toolbar in [Design view](#) of a form or report only in a [Microsoft Access database](#), not a [Microsoft Access project](#).

1. Switch to the appropriate view, and make sure the toolbar you want to put the **Chart** button on is displayed.
2. Right-click anywhere on the toolbar, and then click **Customize** on the shortcut menu.
3. Click the **Commands** tab.
4. In the **Categories** box, click **Toolbox**.
5. Drag the **Chart**  button from the **Commands** box to the toolbar on which you want it to appear.

3. On the form or report, click where you want to place the chart.
4. Follow the instructions in the Chart Wizard, which creates the chart based on available tables or other [record sources](#) and the fields you select.

Note At least one of the fields must have a [data type](#) that stores only numbers, such as Currency.

5. Switch to [Form view](#) or [Print Preview](#) to see the chart with current data.

Note In a [Microsoft Access project](#), you can't use the Chart Wizard, but you can add a chart by inserting the **Microsoft Graph Chart** object (**Insert** menu, **Object** command).

 [Create a record-bound chart from a global chart by linking it to a specific record](#)

1. Open the form or report in [Design view](#).
2. Make sure the chart is selected, and then click **Properties** .
3. In the **LinkChildFields** and **LinkMasterFields** property boxes, type the name of a field that appears in the **RowSource** property box and uniquely identifies each record, such as a [primary key](#).

The values in these property boxes must match record for record the values in a field in the underlying table or other [record source](#) for the form or report. For example, suppose the form's underlying table includes the ProductID field. You want to link your chart on this field, so you type **ProductID** in the **LinkChildFields** and **LinkMasterFields** property boxes.

4. To see whether your chart displays only the data for the current record, display the form in [Form view](#), or look at the report in [Print Preview](#).

 [Add an existing chart](#)

Use this procedure to add an existing chart that was created in another [Microsoft Access database](#) or another program. When you do this, you will not be able to change the chart by changing data in your database.

1. Open the form or report in [Design view](#).

2. Click **Unbound Object Frame**  in the [toolbox](#).
3. On the form or report, click where you want to place the chart.
4. In the **Insert Object** dialog box, click **Create From File**, and then specify a path to the file. If you don't know the path, click **Browse**.
5. If you want to link rather than [embed](#) the chart, select the **Link** check box.
6. Select the **Display As Icon** check box if you want the chart to appear as an icon.

[Show All](#)

About making changes to a chart in a form or report

When you edit a chart, you can either make changes to the look of the chart or the data used to create it.

To change the look of the chart, you can change the size of the chart on the form or report or you can change the appearance of the chart with Microsoft Graph. For example, with Graph, you can change the location of the parts of the chart (such as the legend), the text format, the colors and patterns, and the axis scaling.

To change the data used to create the chart, you can modify the underlying table or other [record source](#), or the query in the **RowSource** property.

When you use the Chart Wizard to create a chart that displays data from Microsoft Access, you choose the table or other record source that includes the fields that are displayed in the chart. This underlying table or other record source contains the data, such as sales figures or customer lists. After you have finished making selections in the Chart Wizard, Access creates a query in the **RowSource** property, which is based on the underlying table or other record source. The query in the **RowSource** property reflects the choices you made in the Chart Wizard to restrict the data, such as a range of dates or orders from a specific region.

For example, if you notice that the sales figures in your chart are wrong, the problem may be in the data stored in the underlying table, or in the design of the underlying record source. In this case, correct the underlying table or other record source. However, if you left out a field that you want to appear in your chart, either you can use the Chart Wizard to create the chart again or you can edit the query in the **RowSource** property.

Note In a [Microsoft Access project](#), you can't use the Chart Wizard, but you can add a chart by inserting the **Microsoft Graph Chart** object (**Insert** menu, **Object** command).

[Show All](#)

Change the data for a chart in a form or report

[Change the data in the underlying table or other record source](#)

1. Do one of the following:

To change the data in an underlying table, open the table in [Datasheet view](#).

To change the data in an underlying [record source](#) that is not a table, open the record source in [Design view](#).

2. Edit the data.
3. Open the form in [Form view](#) or report in [Print Preview](#) to see the chart with current data.

[Change the underlying query of the chart](#)

1. Open the form or report in [Design view](#).
2. Make sure the chart is selected, and then click **Properties** .
3. Click in the **RowSource** property box, and then click **Build**  to the right of the **RowSource** property box to open the Query Builder.

The Query Builder displays the RowSource query in Design view.

4. Change the query.
5. To preview your changes, click **Datasheet** on the **View** menu.
6. Close the Query Builder.
7. Open the form in [Form view](#) or report in [Print Preview](#) to see the chart with current data.

[Show All](#)

Change the look of a chart in a form or report

[Change the size of a chart](#)

1. Open the form or report in [Design view](#).
2. Click the chart.
3. Move the mouse pointer over any of the [sizing handles](#) until the pointer becomes a double-headed arrow.
4. Drag the handle to where you want the edge of the object to be, and then release the mouse button.

[Change the appearance of a chart](#)

1. Open the form or report in [Design view](#).
2. Double-click the chart to open Microsoft Graph.
3. Make the changes you want.

For information about using Graph, click **Microsoft Graph Help** on the **Help** menu.

4. When you finish editing, click **Exit** on the Graph **File** menu.
5. When the application prompts you to update the object, click **Yes**.

Note Although you can change the data that appears in your chart by editing the datasheet in Graph, changes to the data as well as some formatting changes will be overwritten by Microsoft Access. To change the data that is displayed in your chart, you should change the underlying table or other [record source](#), or the query in the **RowSource** property.

[Show All](#)

Add the Chart button to a toolbar

Note You can add the **Chart** button to a toolbar in [Design view](#) of a form or report only in a [Microsoft Access database](#), not a [Microsoft Access project](#).

1. Switch to the appropriate view, and make sure the toolbar you want to put the **Chart** button on is displayed.
2. Right-click anywhere on the toolbar, and then click **Customize** on the shortcut menu.
3. Click the **Commands** tab.
4. In the **Categories** box, click **Toolbox**.
5. Drag the **Chart**  button from the **Commands** box to the toolbar on which you want it to appear.

[Show All](#)

Save disk space with a chart in a form or report

If you need to save disk space, you can convert the chart to a picture, as long as the chart isn't a record-bound chart. Once you do this, you can no longer edit the chart.

1. Open the form or report in [Design view](#).
2. Click the chart.
3. On the **Format** menu, point to **Change To**, and then click **Image**.

[Show All](#)

Troubleshoot charts in a form or report

[I change the chart, but the changes aren't retained.](#)

A chart is made up of information from the following places, in order of precedence:

- The data in the underlying table or other [record source](#)
- The chart's **RowSource** property
- The Microsoft Graph application

Microsoft Access regenerates the chart each time you print or preview the chart, or switch to [Form view](#). Therefore, changes that you make using Graph might be overwritten by the underlying data or the **RowSource** property.

Keep the following in mind when you change your chart:

- Changes to the underlying data (such as the format of the data) have precedence over other changes. However, if the **RowSource** property setting contains a [calculated field](#), such as =Format([StartDate],"MMM"), the format is determined by the result of the calculation, not by the underlying data.
- Changes in the **RowSource** property, such as changes to the chart's title and labels, can be overwritten by changes in the underlying data.
- Changes you make using Graph, such as changes based on the data, can be overwritten by changes you make in the underlying table or other record source or in the **RowSource** property.

[I get the message "Can't bind name name" when trying to display a chart.](#)

You may encounter this message if you didn't explicitly declare in the **Query Parameters** dialog box the parameters used for a [crosstab query](#) or a query that a crosstab query or chart is based on. To resolve this, do the following:

1. In the query that contains the parameter, specify the parameter and its data type in the **Query Parameters** dialog box.

[How?](#)

1. On the **Query** menu, click **Parameters**.
2. In the first **Parameter** cell, type the first prompt you entered in the query [design grid](#).
3. In the **Data Type** cell to the right, click the appropriate [data type](#) according to the following guidelines:

If the parameter field data type is

Click this data type in the Data Type cell

Currency,

Date/Time, Memo, Currency, Date/Time, Memo, OLE Object, Text, OLE Object, Text, and Yes/No and Yes/No

Number

Byte, Single, Double, Integer, Long Integer, and Replication ID. These data types correspond to the five **FieldSize** property settings of the Number data type.

Unknown

Value. This is a generic data type that accepts any type of data.

Binary

Binary. Used with parameter queries directed to linked tables that do recognize this data type (Microsoft Access does not recognize this data type).

4. Repeat Steps 2 and 3 for each parameter you want to specify a data type for.

2. In the same query, set the **ColumnHeadings** property.

[How?](#)

You can change the order or limit the column headings in a [crosstab query](#).

For example, in column headings containing the months of the year, you can display the months chronologically rather than alphabetically. Or, you can limit the columns to just January through June.

1. Open the crosstab query in [Design view](#).
2. Click the background of query Design view, outside the [design grid](#) and the [field lists](#).
3. On the toolbar, click **Properties**  to display the query's property sheet.
4. In the **ColumnHeadings** property box, enter the column headings you want to display, in the order in which you want to display them. Between the column headings, type a comma or the list separator for your country/region. (To find the list separator for your country/region, see the regional settings in Microsoft Windows Control Panel.)

The column headings you enter must exactly match the column headings in the query datasheet. For example, if a column heading in the datasheet is "USA," you must enter a column heading of "USA"—not "US." (After you press ENTER or move the pointer to a different location, Microsoft Access places quotation marks around each heading.)

5. To view the query's results, click **View**  on the toolbar.

Notes

- If you run a crosstab query often, or if you use it as the basis for a form, report, or [data access page](#), you can speed up the query by using the preceding procedure to specify fixed column headings.
- If you frequently use the same column headings in different queries, consider creating a table with one [Text](#) field to store the column headings. Then open the table and copy the headings into the **ColumnHeadings** property box as needed.

 [The chart appears dimmed or blank.](#)

The **RowSource** property for a chart may be invalid.

1. Open the form or report in [Design view](#).

2. Make sure the chart is selected, and then click **Properties** .
3. Make sure the **RowSource** property box contains a valid value.
4. Switch to [Form view](#) or [Print Preview](#) to view the changes.

If the chart is record-bound, the query in the chart's **RowSource** property can't contain a field that matches the fields in the **LinkChildFields** and **LinkMasterFields** properties of the form or report.

[I can't change the format of the labels on my chart.](#)

If you use Microsoft Graph to change the format of labels on your chart's axes, the changes are overwritten when the chart is updated. To make these changes permanent, you must modify the [query](#) specified by the chart's **RowSource** property.

1. Open the form or report in [Design view](#).
2. Make sure the chart is selected, and then click **Properties** .
3. Click **Build**  to the right of the **RowSource** property box.
4. In the query, replace the name of the field you want to format with the field name and the **Format** function.

For example in a [Microsoft Access database](#), you can change it to `Format([OrderDate],"YY")` to change how the date appears on your chart.

5. Close the query and save the changes.
6. Switch to [Form view](#) or [Print Preview](#) to view the changes.

Note Not all Access custom formats work with Graph.

[I can't put the columns on my chart in the order I want.](#)

To make your data appear in a different order, you can modify the query specified by the chart's **RowSource** property.

1. Open the form or report in [Design view](#).
2. Make sure the chart is selected, and then click **Properties** .
3. Click **Build**  to the right of the **RowSource** property box.
4. In the query, put the fields you want sorted, left to right, in the order you want them sorted.

5. Close the query and save the changes.
6. Switch to [Form view](#) or [Print Preview](#) to view the changes.

[I see sample data instead of my data after creating a chart.](#)

In form or report [Design view](#), the chart shows sample data when you first create it. To see current data, switch to [Form view](#) or [Print Preview](#), or close the form or report and then open it.

[I can't create a radar, surface, or combination chart.](#)

You can't create a radar, surface, or combination chart by using the Chart Wizard. However, in Microsoft Graph you can change the chart you created with the Chart Wizard to any chart type, including a radar, surface, or combination chart.

[Show All](#)

Apply or change conditional formatting on a control

1. Open the form in [Design view](#) or [Form view](#), or the report in Design view.

Note You cannot apply conditional formatting to a control on a data access page.

2. Do one of the following:
 - If you are in Design view, click the control for which you want to apply conditional formatting.
 - If you are in Form view, make sure the control has [focus](#).
3. On the **Format** menu, click **Conditional Formatting**.
4. Do one or more of the following:



[Change the format of a control when it meets certain criteria](#)

1. Do one of the following:

To use values in the selected control as the formatting criteria, click **Field Value Is** in the first box, click the comparison phrase in the second box, and then type a value in the third (and fourth) box. You can enter a constant value or an expression.

To use an expression as the formatting criteria (to evaluate data or a condition other than the values in selected controls), click **Expression Is** in the first box, and then enter the expression in the second box. The expression must return a logical value of TRUE or FALSE.

2. Select the font style, color, or other formatting that you want to apply. Microsoft Access applies the selected formatting only if the control value meets the condition, or the expression returns a value of TRUE.
3. To add another condition, click **Add**, and then repeat steps 1 and 2.



[Change the color of the control that has focus](#)

1. In the **Condition x** area of the **Conditional Formatting** dialog box, click **Field Has Focus** in the first box.
2. Select the font style, color, or other formatting that you want the field to have when it has focus.

[Change, add, or remove a condition](#)

- To change an existing condition, select the font style, color, or other formatting that you want to apply.
 - To add a new condition, click **Add**.
 - To remove one or more conditions, click **Delete**, and then select the check box for the condition or conditions you want to delete.
5. Click **OK**.

Notes

- When you open the **Conditional Formatting** dialog box, it displays the current default formatting for the selected control, including colors, font styles, and any expressions that you may have defined for the field.
- You can specify up to three conditions for each control.
- Conditional formatting is not preserved when you output or [export](#) a form or a report to another file format.
- If none of the specified conditions are true, the control keeps its existing formatting. If more than one specified condition is true, Access applies only the formatting of the first true condition.

[Show All](#)

Copy the conditional formatting of a control

1. Open the form or the report in [Design view](#).
2. Select the [control](#) that has the [conditional formatting](#) you want to copy.
3. Click **Format Painter**  on the toolbar.
4. Click the control you want to copy the formatting to.

[Show All](#)

About applying conditional formatting to a control

If a [control](#) on a form or report contains values that you want to monitor, apply [conditional formatting](#) to the control to make it easier to spot. For example, you can set conditional formatting so that if an employee's salary exceeds \$100,000 the field's background turns red. Or you can set formatting so that text in the City field is green and italic if a customer resides in Seattle.

You can set conditional formatting based on the value in a control, an arbitrary expression referencing another control, a user-defined [Visual Basic for Applications function](#), or the control with [focus](#). You can change the color of the text, make the text bold, italic, or underlined, or enable or disable a control when it meets or doesn't meet criteria you specify.

If the value of the control changes and no longer meets the specified condition, Microsoft Access returns to the default formatting for the control. Conditional formatting remains applied to the controls until you remove the formatting, even if none of the conditions are met and the specified control formatting is not displayed.

You cannot use wildcards— such as the asterisk (*), question mark (?), or any other symbol— in criteria as substitutions for text or number characters.

Note You cannot apply conditional formatting to a control on a data access page.

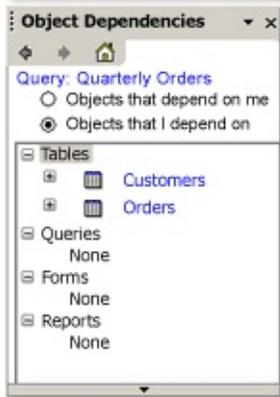
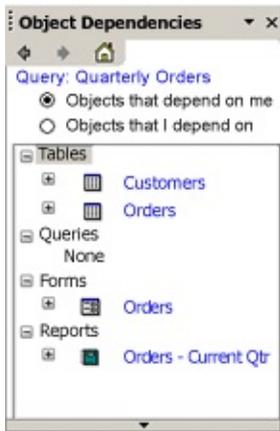
[Show All](#)

About object dependencies (MDB)

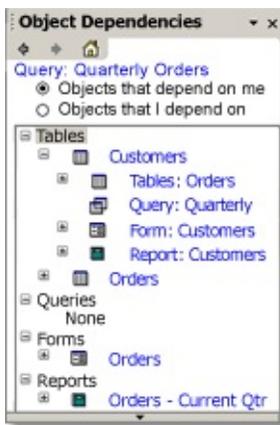
Note The information in this topic applies only to a Microsoft Access database (.mdb).

In Microsoft Office Access 2003, you can view information on dependencies between [database objects](#). Viewing a list of objects that use a specific object helps maintain a database over time and avoid errors related to missing [record sources](#). For example, the Quarterly Orders query in the Sales database is no longer needed, but before deleting it, you might want to find out which other objects in the database use the query. Then, you could either modify the dependent objects to remove dependencies, or delete the dependent objects, before deleting the Quarterly Orders query. Viewing a complete list of dependent objects helps you save time and minimize errors.

In addition to viewing the list of objects that use the selected object, you can also view the objects that are being used by the selected object. For example, the following illustration shows the objects use the Quarterly Orders query, and also the tables and queries that the query uses.



You can click on the expand (+) icon next to an object to see dependency information for that object.



You can navigate through up to four levels.

Access 2003 generates dependency information by searching the name maps maintained by the [Name AutoCorrect](#) feature. If tracking name AutoCorrect information is turned off, you cannot view dependency information.

Note Tracking name AutoCorrect information must be enabled to view dependency information but performing name Autocorrect need not be enabled.

If any object is open, it must be saved and closed so that Access can generate accurate information.

The dependency information is generated only for [tables](#), [queries](#), [forms](#), and [reports](#) in an Access database. It is not available for the following:

- [data access pages](#), [macro](#), and [modules](#) in a database.
- objects not supported by the name AutoCorrect feature, such as action queries and SQL-specific queries, including union queries, data-definition queries, pass-through queries, and subqueries.
- Access projects.

Note Nested queries are partially supported. Access can generate dependency information only for the outermost query. Also, the source tables and queries of a query's [subdatasheet](#) and [lookup fields](#) are not listed as objects that the query depends on.

Dependency information is also not available if you don't have permissions to open an object in [Design view](#).

For each type of object, the following table lists the types of objects it can depend on, and the nature of dependency.

Selected Depends object	on	Nature of dependency
		A relationship is defined between the objects.
Table	A table or query	Selected table includes the table or query as a subdatasheet. Fields in the selected table look up values in the other object.
Table	A form	Fields in the selected table look up values in the form.
Query	A table or query	The selected query is bound to the table or query. A relationship is defined between the objects.

Query	A form	<p>The form is referred to in the query's SQL statement.</p> <p>The selected form or one of its subforms is bound to the table or query.</p>
Form	A table or query	<p>One or more list boxes and combo boxes in the form get data from the table or query.</p> <p>Note Dependencies that arise from a chart on a form getting data from a table or query is not included. You will have to manually identify such dependencies.</p> <p>The selected form includes the other form as a subform.</p>
Form	A form	<p>The record source of the selected form or the row source of a list box or combo box in the selected form uses a SQL statement that refers to the other form.</p> <p>The selected report or one of its subreports is bound to the table or query.</p> <p>One or more list boxes and combo boxes in the report get data from the table or query.</p>
Report	A table or query	<p>Note Dependencies that arise from a chart on a report getting data from a table or query is not included. You will have to manually identify such dependencies.</p> <p>The record source of the selected form or the row source of a list box or combo box in the selected form uses a SQL statement that refers to the other form.</p>
Report	A form	<p>The record source of the selected report or the row source of a list box or combo box in the selected report uses a SQL statement that refers to the form.</p>

The selected report includes the form as a subform.

Report A report The selected report includes the other report as a subreport.

[Show All](#)

About ignored objects (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When Microsoft Office Access 2003 generates dependency information, it ignores one or more objects for the following reasons:

[Insufficient permission](#)

You don't have permission to open an object in [Design view](#). Access 2003 needs to open, save, and close an object to generate dependency information for it. To get permissions for one or more objects in the database, contact the owner of the database.

[Unsupported objects](#)

Access 2003 uses name maps created by the name AutoCorrect feature to generate dependency information. Objects such as action queries and SQL-specific queries, including union queries, data-definition queries, pass-through queries, and subqueries, do not support the [Name AutoCorrect](#) feature, so dependency information for such objects cannot be generated. Due to this limitation, the dependency information you are viewing may not be complete.

View dependency information (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can view dependency information for an open object, or an object that is selected in the Database window.

Tracking name AutoCorrect information must be turned on to view dependency information. The dependency information is generated only for tables, queries, forms, and reports in an Access database. It is not available for the following - Access projects, data access pages, macros and modules in a database, and objects not supported by the name AutoCorrect feature, such as action queries and SQL-specific queries, including union queries, data-definition queries, pass-through queries, and subqueries. Dependency information is also not available if you don't have permissions to open an object in Design view.

- In the **View** menu, click **Object Dependencies**.

The dependency pane shows the list of objects that use the selected object.

- To view the list of objects that are being used by the selected object, click **Objects that I depend on** on top of the pane.

The dependency pane shows the list of objects that are being used by the selected object.

- To view dependency information for an object listed in the pane, simply click on the expand icon (+) next to it.

Note Hidden objects will not be listed, unless the **Hidden Objects** option on the **View** tab in the **Options** dialog box is selected.

[Show All](#)

About using Name AutoCorrect to fix errors caused by renaming (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

In Microsoft Access 97 and previous versions, if you rename a field in a table, the objects based on that field no longer display the data from that field. In Access 2000 or later, you can make Access track name changes, or automatically correct common side effects that occur when you rename forms, reports, tables, queries, fields, text boxes or other controls in a [Microsoft Access database](#). This feature is called [Name AutoCorrect](#).

You can enable name AutoCorrect in three levels:

- Track name changes.

When you choose to track name changes, Access simply keeps track of the name changes in the database. It does not fix the errors caused by renaming. The tracking information is stored in name maps. When you enable tracking for the first time, Access creates name maps for each existing database object, keeps it up-to-date. The maps are used to fix errors caused by renaming, as well as to generate object dependency information.

- Track changes and automatically fix errors caused by renaming.

After enabling tracking, you can choose to have Access automatically fix name changes as and when they are made. The name maps created when you enabled tracking are used to fix the errors.

- Track changes, fix errors, and maintain a log of changes made by name AutoCorrect.

Access will maintain a log of changes made to object names. Each change will appear as a record in the **Name AutoCorrect Log** table.

The name AutoCorrect settings are available on the **General** tab of the **Options** dialog box.

Name AutoCorrect and object dependency information

Microsoft Office Access 2003 makes it possible to view dependency information for tables, queries, forms, and reports. You can view the list of objects that depend on a selected table or form, and also view the objects that the table or form depends on. The dependency information is generated from name maps that are created when you enable name AutoCorrect. The name map of an object is updated each time you save and close it.

[Enabling tracking name changes](#)

When you enable tracking name changes, Access does the following:

1. Attempts to get exclusive access to the database.
2. Searches for objects that are open.
3. Prompts and closes open objects.
4. Updates name maps by opening, saving, and closing all tables, queries, forms, and reports.

This step can take several minutes. You can press Esc to cancel this step. However, the name maps of some objects will be outdated, and you will not be able to view the dependency information of such objects, until you enable the name AutoCorrect feature again.

Note If you interrupt Access when it is updating the name maps, Access does not turn off tracking. The Track name AutoCorrect option will be left selected even though some name maps are outdated. To restart the process of updating name maps, clear the option, click **Apply**, and then select it again. Or, you can manually update an object's name map by opening it in Design view and saving it.

- Access returns the database to whatever state it was in before it was promoted to Exclusive.

Tracking name changes is not automatically turned on when you convert databases created using Access 97 or earlier. When you create a database using

Access 2000 or later, tracking is turned on by default.

[Turning off tracking name changes](#)

When you turn off tracking name changes, Access does the following:

1. Attempts to get exclusive access to the database.
2. Searches for objects that are open.
3. Prompts and closes open objects.
4. Removes name maps from all tables, queries, forms, and reports.
5. Access returns the database to whatever state it was in before it was promoted to Exclusive.

If Access fails to complete the process of tracking changes or removing name maps due to errors or lack of sufficient permissions, it will log the error in the Name AutoCorrect Save Failures table.

When tracking name changes is enabled for a database, you might notice an increase in the database size. Removing name maps reduces the size of the database.

Note If you encounter an error while trying to dependency information of an object, make sure you have permission to save objects. Also, clear the **Track name AutoCorrect info** check box on the **General** tab of the **Options** dialog box, click **Apply**, and then select the check box again.

[Show All](#)

Automatically repair errors caused by renaming (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. On the **Tools** menu, click **Options**.
2. On the **General** tab, do one or more of the following:
 - To have Microsoft Access maintain the information it needs to run [Name AutoCorrect](#), but not take any action, select the **Track name AutoCorrect info** check box.
 - To have Access run Name AutoCorrect as name changes occur in the database, select the **Perform name AutoCorrect** check box.
 - To have Access keep a log of the changes it makes each time it runs Name AutoCorrect, select the **Log name AutoCorrect changes** check box. Access saves the changes in a table called Name AutoCorrect Log. Each change is a record in the Name AutoCorrect Log table.
3. To enable Name AutoCorrect for a table, query, form or report that already exists in the Access database, open it in [Design view](#), save it, and close it.

Notes

- If you want to view dependency information for a table, query, form, or report, you must select the **Track name AutoCorrect info** check box. This option creates or updates the name maps that are necessary for generating dependency information.
- If the **Track name AutoCorrect info** check box is selected, and you encounter errors when trying to view dependency information, make sure you have permission to save and close objects. Also try clearing the check box, clicking **Apply**, and then reselecting the **Track name AutoCorrect info** check box.

[Show All](#)

Troubleshoot Name AutoCorrect (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

 [Name AutoCorrect doesn't fix errors after I have renamed a database object or a field.](#)

Name AutoCorrect doesn't fix references in:

- A [Microsoft Access project](#).
- A [replicated](#) database. When you replicate your database, Access turns Name AutoCorrect off. You can't turn on Name AutoCorrect in a database [Design Master](#) or [replica](#).
- [Pass-through](#), [data-definition](#), and [union](#) queries.
- [Data access pages](#), [macros](#), and [modules](#).
- Desktop shortcuts that open a particular database object.
- An invalid SQL statement.
- Microsoft Visual Basic code.
- A linked table when the table or fields have been renamed in the back end of a [front-end/back-end database](#).
- The **Filter** or **OrderBy** properties of tables, queries, forms, and reports.

In addition, Name AutoCorrect doesn't fix references to macros in a toolbar or menu, including menu and toolbar properties and [event properties](#) that are set to macro names.

If you import or create a database object when Name AutoCorrect is not turned on, Access doesn't have enough information to enable NameAutoCorrect for that object. To enable Name AutoCorrect for the object, do the following:

1. On the **Tools** menu, click **Options**, and then click the **General** tab.
2. Select the **Track name AutoCorrect info** and **Perform name AutoCorrect** check boxes.

If you want a log of the changes that Access makes to the database when it repairs naming errors, select the **Log name AutoCorrect changes** check box as well.

3. Open the database object in [Design view](#), save it, and close it.

[Name AutoCorrect doesn't work in a converted database.](#)

Microsoft [Name AutoCorrect](#) is on by default in databases that were created with Microsoft Access 2000 or later. Name AutoCorrect is automatically turned on when you convert databases created using Access 97 or earlier to Access 2003 . If Name AutoCorrect is not enabled in a converted database, do the following:

1. On the **Tools** menu, click **Options**, and then click the **General** tab.
2. Select the **Track name AutoCorrect info** and **Perform name AutoCorrect** check boxes.

If you want a log of the changes that Access makes to the database when it repairs naming errors, select the **Log name AutoCorrect changes** check box as well.

[Name AutoCorrect doesn't work for imported objects.](#)

When you import an object into an Access 2000 or later database that has tracking enabled, the tracking information is not automatically updated for the imported objects. You must save and close the objects to enable tracking. If you import objects into an Office Access 2003 database, the tracking information for the imported objects are automatically updated.

[Show All](#)

Troubleshoot viewing dependency information (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Microsoft Office Access 2003 generates dependency information by searching the name maps maintained by the Name AutoCorrect feature. You cannot view dependency information in the following situations:

[Tracking name AutoCorrect information is turned off.](#)

If this feature is turned off, Access 2003 cannot search the name maps, so it cannot generate dependency information. Turn on this feature to create name maps for the objects in the database.

- Select **Track name AutoCorrect info** check box on the **General** tab in the **Options** dialog box.

Note To have Access 2003 run name AutoCorrect as name changes occur in the database, select the **Perform name AutoCorrect** check box.

[Name maps are missing.](#)

If one or more name maps are missing, the dependency information generated by Access 2003 will not be complete.

Do one of the following:

- Turn on name AutoCorrect if it is disabled by selecting **Track name AutoCorrect info** check box on the **General** tab in the **Options** dialog box.
- If name maps are missing when name AutoCorrect is already turned on, it could be because the name map creation process was interrupted. To restart the process of creating name maps, clear the **Track name AutoCorrect info** check box, click **Apply**, then select the check box again.

[Name maps cannot be created or updated.](#)

To create or update a name map, Access 2003 must open, save, and close each object in the database. If an object is already open but has unsaved changes, or is read-only, or if you don't have permissions to open and save an object, name maps will not be created or updated.

1. Make sure you have permissions to save objects.
2. Close all objects.
3. Restart the process of creating name maps. Clear the **Track name AutoCorrect info** check box, click **Apply**, then select the check box again.

[Dependency information for new unsaved objects cannot be generated.](#)

An object must be saved at least once before you can view its dependency information. Access creates a name map for the object when it is saved. If the object that is currently open in Design view is new, and has not been saved yet, you will see an error message prompting you to save the object. Save the object, and then try displaying the dependency information.

[Show All](#)

About error checking in forms and reports

In Microsoft Office Access 2003, you can enable automatic error checking for common errors in [form](#) and [reports](#). Error checking points out errors in a form or report, and provides you with options for correcting them.

The following illustration shows the BirthDate label control in [Design view](#). Error checking for errors related to shortcut keys is turned on, so you see an error indicator on the top left corner of the [control](#). The control has a duplicate shortcut key assigned to it.

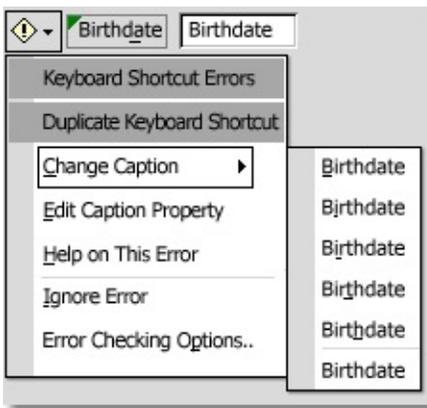


If page orientation is set to right-to-left, you will see error indicators on the top right corner.

If you select the control with an error indicator, the **Error Checking Options** button appears.



If you rest your mouse pointer on the **Error Checking Options** button and then click the arrow in the button, a menu displays a set of options to correct or ignore the error.



Types of errors

The following table lists the types of errors for which you can enable error checking.

Category	Description	Applies To
Unassociated label and control	<p>You select a label and a control, such as a text box or list box, that are not associated with each other.</p> <p>The error indicator will not appear; instead, the Trace Error button will appear next to the selected control. - You will see the Trace Error button even if the selected label or control is associated with another control.</p>	Forms and reports
New unassociated labels	<p>You add a label to a form or report that is not associated with any other control.</p>	Forms and Reports
Keyboard shortcut errors	<p>You select a control with an invalid shortcut key, such as:</p> <ul style="list-style-type: none">• An unassociated label has a shortcut key associated with it.• A label or a button has an duplicate shortcut key assigned to it.• A label or a button has a space character as its shortcut key.	Forms only
Invalid control properties	<p>You select a control with one or more properties set to invalid values, such as:</p> <ul style="list-style-type: none">• The string specified in the Control Source property is not a valid expression or field name.• The expression specified in the Control Source property refers to the control itself (circular reference).• The expression specified in the Control Source property does not begin with “=”.• The Option Value property is not unique in the	Forms and Reports

[option group](#).

Common report errors	The current report either has an invalid sorting and grouping definition, or is wider than the selected paper size.	Reports only
----------------------	---	--------------

[Resolving multiple errors](#)

If a control has multiple errors, you will see the error indicator until you correct or ignore all the errors. For example, the following errors, listed from highest to lowest priority, can occur on the BirthDate label at the same time:

- Space character as a shortcut key.
- Unassociated label with shortcut key.
- Duplicate shortcut key.
- New unassociated label.

For example, if the BirthDate label is not associated with any control and it has a space character specified as a shortcut key, you will first see the error indicator for the invalid shortcut key. After you correct or ignore the error, you can click the error indicator again to associate it with a control, or you can ignore the error, and so on.

[Ignoring an error](#)

If you want to hide the error indicator and the **Error Checking Options** button without fixing the error, you can choose to ignore the error by clicking **Ignore Error** on the **Error Checking Options** menu. The error indicator and the **Error Checking Options** button will disappear until you open the form or report again.

[Turning off error checking](#)

To turn off error checking for one or more types of errors, you can click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, you can specify the settings you want. You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Turn error checking on or off

1. On the **Tools** menu, click **Options** and then click the **Error Checking** tab.
2. Under **Settings**, select the **Enable error checking** check box.
3. Under **Form/Report Design Rules**, select or clear the check boxes you want.

Note To change the color of the error indicator , select the color you want from the **Error indicator color** list.

Circular reference

The **Control Source** property of the selected control is an expression that includes a reference to the control. Circular reference will cause the expression to fail.

You can do one or more of the following:

- To edit the control's **Name** property in the property sheet, click **Edit Control Name** on the **Error Checking Options**  menu.

Note If Name AutoCorrect is turned on, changing the name of the control will automatically update the control's **Control Source** property. The circular reference error will still exist, so to remove the error, edit the control's **Control Source** property to reverse the automatic update.

- To edit the **Control Source** property of the control, click **Edit the Control's Control Source Property** on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for circular-reference errors, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Invalid control properties** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Duplicate keyboard shortcut

The selected label or button has a shortcut key that has already been assigned to another control on the form.

You can do one or more of the following:

- To choose from a list of suggested captions, point to **Change Caption** on the **Error Checking Options**  menu and select the caption you want.
- To edit the **Caption** property in the property sheet, click **Edit Caption Property** on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for labels with duplicate shortcut keys, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Keyboard shortcut errors** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Duplicate option value

The selected control has the same option value as one or more other controls in the option group.

You can do one or more of the following:

- To edit the control's **Option Value** property in the property sheet, click **Edit Option Value** on the **Error Checking Options**  menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for duplicate option values, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Invalid control properties** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Expressions must begin with an equal sign (=)

The **Control Source** property of the selected control is set to an expression, but the expression does not begin with an equal sign (=).

You can do one or more of the following:

- To edit the control's **Control Source** property in the property sheet, click **Edit the Control's Control Source Property** on the **Error Checking Options**  menu.
- To edit the **Record Source** property of the form or report in the property sheet, click **Edit the Form's(or Report's) Record Source Property** on the **Trace Error** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for incorrect expression syntax, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Invalid control properties** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Invalid sorting and grouping

The report's sorting and grouping information contains an invalid field name.

You can do one or more of the following:

- To edit the report's sorting and grouping information, click **Edit Sorting and Grouping** on the **Error Checking Options**  menu.

Note If the report has multiple errors, point to **Invalid Sorting and Grouping** on the **Error Checking Options** menu, and then click **Edit Sorting and Grouping**.

- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for invalid sorting and grouping information, click **Error Checking Options** on the **Trace Error** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Common report errors** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

New unassociated label

You added a label to a form or report, and it is not associated with any control.

You can do one or more of the following:

- To associate the label with a control, click **Associate Label with a Control** on the **Error Checking Options**  menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for new unassociated labels, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **New unassociated labels** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

No such field in the field list

The control source property of the selected control is not a valid expression or field name.

You can do one or more of the following:

- To edit the **Control Source** property in the property sheet, click **Edit the Control's Control Source Property** on the **Error Checking Options**  menu.
- To edit the **Record Source** property of the form or report, click **Edit the Form's (or Report's) Record Source Property** on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for controls with an invalid control source, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Invalid control properties** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Report width is greater than page width

The width of the report is greater than the width of the paper specified in the **Page Setup** dialog box. Your printed report might not appear as expected.

You can do one or more of the following:

- To edit the report margins, click **Edit Margins** on the **Error Checking Options**  menu.

Note If the report has multiple errors, point to **Report Width Greater than Page Width** on the **Error Checking Options** menu, and then click **Edit the Margins**.

- To eliminate the white space on the right side (or left side, if the report's orientation is set to right-to-left) of the report, click **Remove Extra Report Space** on the **Error Checking Options** menu.
- If you want to select the control that is the farthest to the right (or left, if the report's orientation is set to right-to-left), click **Select the Control Farthest to the Right** (or **Left**) on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for report width, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Common report errors** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Spacebar is not a valid keyboard shortcut

The selected label has an invalid shortcut key assigned to it. The ampersand (&) in the caption text is followed by a space.

You can do one or more of the following:

- To choose from a list of suggested captions, point to **Change Caption** on the **Error Checking Options**  menu and select the caption you want.
- To edit the **Caption** property in the property sheet, click **Edit Caption Property** on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu.
- To turn off error checking for labels with invalid shortcut keys, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Keyboard shortcut errors** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

Unassociated label and control

The selected label and control are not associated with each other.

You can do one or more of the following:

- To associate the label with the selected control, click **Associate** on the **Error Checking Options**  menu.
- To ignore this error and hide the **Error Checking Options** button, click **Dismiss Error** on the **Error Checking Options** menu.
- To turn off error checking for unassociated labels and controls, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Unassociated label and control** check box.

Unassociated label with keyboard shortcut

The selected label has a shortcut key assigned to it, but is not associated with any control. This implies that when the user presses the shortcut key, nothing will happen.

You can do one or more of the following:

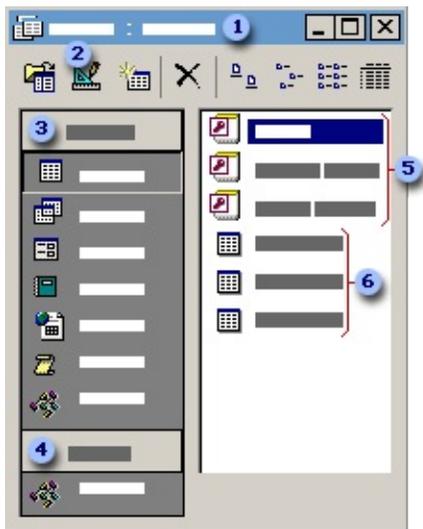
- To associate the label with a control, click **Associate Label with a Control** on the **Error Checking Options**  menu.
- To remove the shortcut key, click **Remove Keyboard Shortcut** on the **Error Checking Options** menu.
- To ignore this error and hide the **Error Checking Options** button, click **Ignore Error** on the **Error Checking Options** menu. If the label has multiple errors, you will see the error indicator until you correct or ignore all the errors.
- To turn off error checking for unassociated labels with shortcut keys, click **Error Checking Options** on the **Error Checking Options** menu. On the **Error Checking** tab, under **Form/Report Design Rules**, clear the **Keyboard shortcut errors** check box.

Note You can also access the **Error Checking** tab by clicking **Options** on the **Tools** menu, and then clicking the **Error Checking** tab.

[Show All](#)

About ways to use the Database window

When you create or open a Microsoft [Access file](#), the Database window appears. The Database window is the command center of your Access file. From here, you can create and use any object in your Access database or Access project.



- 1 The title bar of the Database window shows the name and file format of the database.
- 2 On the Database window toolbar, use the **Open** button to work with existing objects, use the **Design** button to modify existing objects, or use the **New** button to create new objects.
- 3 Under **Objects**, click one of the object types, such as **Tables** or **Forms**, to show the list of objects of that type.
- 4 A list of groups of database objects appears under **Groups**. You can add objects of different types to a group, which consists of shortcuts to the database objects that belong to it.
- 5 You can use the new object shortcuts at the top of the object list to create new database objects.

6 The list of database objects changes according to which object type you have clicked under **Objects**.

[Show All](#)

Refresh the Database window

In a [multiuser](#) environment, refreshing the [Database window](#) ensures that the list of objects in the Object list is current.

1. Press F11 to switch to the Database window.
2. Press F5 to refresh the Database window.

Note Microsoft Access automatically refreshes the Database window when you open an Access database or Access project, or when you create a new [database object](#) in a Microsoft SQL Server database from the Access project that is connected to it.

[Show All](#)

Change the appearance of objects in the Database window

1. In the [Database window](#), click one of the following buttons on the Database window toolbar: **Large Icons** , **Small Icons** , **List** , or **Details** .
2. To sort the objects, on the **View** menu, point to **Arrange Icons**, and click one of the following: **By Name**, **By Type**, **By Created**, or **By Modified**.

[Show All](#)

Display multiple database objects on the Windows taskbar

If you have Microsoft Internet Explorer Active Desktop installed, you can use the Microsoft Windows taskbar to switch between [database objects](#) that are currently open.

1. On the **Tools** menu, click **Options**, and then click the **View** tab.
2. Select or clear the **Windows in Taskbar** check box. If this box is selected, each open database object will have an icon displayed on the Windows taskbar.

Notes

- To determine whether Microsoft Internet Explorer Active Desktop is installed on your computer, right-click your desktop. If **Active Desktop** appears on the shortcut menu, then it has been installed. If you do not have Microsoft Internet Explorer Active Desktop installed, you can still select the **Windows in Taskbar** check box, but it has no effect.
- Selecting or clearing the **Windows in Taskbar** check box does not affect taskbar switching behavior in other Microsoft Office applications.

[Show All](#)

Troubleshoot converting an Access file

You can convert most previous-version [Microsoft Access files](#) to Access 2000 or Access 2002 - 2003 file format with no difficulty. In some rare cases, however, new features can conflict with existing objects and code.

[Troubleshoot converting any previous-version Access database to Access 2000 or Access 2002 - 2003 file format](#)

[I receive a message that there were compilation errors during the conversion or enabling of the database.](#)

When you convert a Microsoft Access version 2.0 database to Access 2000 or Access 2002 - 2003 file format, the conversion utility converts Access Basic code to Visual Basic for Applications (VBA) code. If any code doesn't successfully compile, you might receive an error message after you have converted the Access database.

In the converted Access database, open a module in Design view and click **Compile <Project Name>** on the **Debug** menu. As Access compiles, it stops at any line of code that contains an error. You can then modify the syntax to resolve the compile error.

- In rare instances, the conversion utility may fail to convert some syntax. For example, it may fail to modify a **DoCmd** statement into a **DoCmd** method. You may also receive compile error messages if your Microsoft Access version 2.0 database contains syntax errors. For this reason, it is a good idea to compile all of your previous-version database modules before you convert them.
- Microsoft Access 2000 or later does not support the [DAO 2.5/3.x](#) compatibility library. If you attempt to convert a database in which the code contains older versions of DAO objects, methods, and properties that depend on the DAO 2.5/3.x compatibility library, you receive a message

that there were compilation errors during the conversion of the database. Before you convert the database, update the code so that it does not depend on the DAO 2.5/3.x compatibility library. If you still receive a message that there were compilation errors during conversion, use the following procedure to correct the references.

1. Open the converted Access database in Access 2002 or later.
 2. Open a [module](#) in the database.
 3. On the **Tools** menu, click **References**.
 4. In the **Available References** box, clear the check boxes next to the missing references to the DAO 2.5/3.x compatibility library.
 5. Select the check box next to the Microsoft DAO 3.6 Object Library.
- Microsoft Access 97 or later supports some new Visual Basic keywords, so you can no longer use these keywords as identifiers. These keywords are: **AddressOf, Decimal, DefDec, Enum, Event, Friend, Implements, RaiseEvent, WithEvents**. When you convert a previous-version Access database to Access 2000 or Access 2002 - 2003 file format, existing identifiers that are the same as a new Visual Basic keyword will cause a compile error. To fix this problem, change existing identifiers so they are not keywords.
 - In Access version 2.0, you could specify a name for a Visual Basic [procedure](#) that was the same as a module name. However, in Microsoft Access 95 or later, a procedure and a module can't have the same name. To fix this problem, rename the procedure or rename the module. Alternatively, you can avoid this error by using a fully qualified name for the procedure in your Visual Basic for Applications (VBA) code.
 - When you convert an Access version 2.0 database to Access 2000 later, the code in the new file is contained in a Visual Basic for Applications project. By default, the name of the VBA project is the same as the name of the Access file. The name of the VBA project is also a reserved word in Visual Basic. If a module or a procedure has the same name as the VBA project, you receive a message that there were compilation errors during conversion.

[I receive the message "Out of Memory" or "Can't create any more class modules" when converting a large database.](#)

A file in Microsoft Access 2000 or Access 2002 - 2003 file format has a limit of 1000 [modules](#), while an Access 97 database has a limit of 1024 modules. The

number of modules in a Microsoft Access file includes forms and reports with the **HasModule** property set to **Yes**. To fix this memory problem, reduce the number of [objects](#) in your Access database. Consider dividing your application into multiple databases. If you have modules with a large amount of code, consider using [library databases](#) to store the code.

[I want to see a log of my conversion errors.](#)

If Microsoft Access encounters errors while converting an [Access file](#), you can view a summary of these errors by opening the Conversion Errors table in the new Access file. The Conversion Errors table contains the following columns:

Object Type The type of [database object](#) in which Access encountered an error, or "Database" if Access encountered an error that is not specific to a particular type of object.

Object Name The name of the object in which Access encountered an error. If Access encounters a compilation error during conversion, however, the name of the module that contains the error is not specified.

Error Description If necessary, you can press SHIFT+F2 to view the entire description of the error.

[I can't convert my Access file by using the **Convert Database/Project** command.](#)

If you can't successfully convert your Microsoft Access file by pointing to **Database Utilities** on the **Tools** menu and then clicking the **Convert Database/Project** command, use the following procedure.

1. In Access 2002 or later, click **New** on the **File** menu and create a new [Access database](#) or [Access project](#).
2. On the **File** menu, point to **Get External Data** and click **Import**.
3. In the **Import** dialog box, select the Access file you want to convert.
4. In the **Import Objects** dialog box, select the [database objects](#) you want to import. If the new Access file is an Access project that you have connected

to a Microsoft SQL Server database, import only the forms, reports, pages, macros, and modules.

Note When you convert an Access file by using this method, you might have to set references by opening a module in Design view, clicking **References** on the **Tools** menu, and selecting the check boxes next to the references that were set in the previous-version Access file.

[Name AutoCorrect doesn't work in a converted Access database.](#)

[Name AutoCorrect](#) is on by default in databases that were created with Microsoft Access 2000 or later. To enable Name AutoCorrect in a converted [Microsoft Access database](#), do the following:

1. On the **Tools** menu, click **Options**, and then click the **General** tab.
2. Select the **Track name AutoCorrect info** and **Perform name AutoCorrect** check boxes.

If you want a log of the changes that Access makes to the database when it repairs naming errors, select the **Log name AutoCorrect changes** check box as well.

3. Open each table, query, form, and report in [Design view](#), save it, and close it.

[The data in a converted non-English Access database is illegible.](#)

When you convert a [Microsoft Access database](#) from Access 97 or earlier to Access 2000 or Access 2002 - 2003 file format, Access uses the sort order to determine which [code page](#) to use when converting the data to [Unicode](#). Access 2000 or later associates the **General** sort order with the Western European code page. Therefore, if you used the previous-version Access database with a non-English operating system and saved it with the General sort order, use one of the following procedures.

- If your operating system is in the same language as the database or supports that language, use the following procedure.
 1. Open the database in the previous non-English version of Access.

2. Change the sort order to match the language of the operating system. For information on how to change the sort order in that version of Access, search Access Help.
 3. Compact the database.
 4. In Access 2002 or later, convert the database.
- If your operating system isn't in the same language as the database or doesn't support that language, or if the language had no sort order in the previous version of Access, use the following procedure.

You might need to use this procedure if the language of the Access database you want to convert was created with an operating system in Vietnamese, Farsi, or a Baltic language (Estonian, Latvian, or Lithuanian).

1. Do one of the following:

- If you are converting an Access 2.0 database, locate the following subkey in the Windows Registry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\4.0\Engines\  
2.x
```

- If you are converting an Access 95 or 97 database, locate the following subkey in the Windows Registry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\4.0\Engines\  
3.x
```

2. In the Jet 2.x or Jet 3.x subkey, create a new entry named ForceCP and do one of the following:

- Set the value of the entry "ANSI" to specify the computer's default code page.
- Specify the code page number for the language you want. For example, specify "1257" for Windows Baltic Rim.

3. Start Access 2002 or later and convert the database.

4. Delete the ForceCP registry entry so that Access reverts to using the sort order of the database to determine the code page.

[Troubleshoot additional issues when converting an Access version 2.0 database to Access 2000 or Access 2002 - 2003 file format](#)

[Time or date values in my query criteria return different results.](#)

When you convert a Microsoft Access version 2.0 database to Access 2000 or Access 2002 - 2003 file format, queries that contain criteria based on specific time values in Date/Time fields may return different results than they do in earlier versions. This behavior may also occur if you link tables from a Microsoft Access version 2.0 database to a Access 2000 or later database. Only the time portion of Date/Time fields is affected.

Queries that contain criteria specifying dates between 1900 and 1929 may also return different results. For example, a date criteria of **#01/01/15#** in Microsoft Access 2.0 and 95 represents January 1, 1915; the same date criteria in Microsoft Access 97 or later represents January 1, 2015. To work around this difference, modify the data in your criteria to specify the century— for example, **#01/01/1915#**.

[My converted Microsoft Access version 2.0 report has different margins.](#)

In a Microsoft Access database that you've converted from Access 2.0 to Access 2000 or Access 2002 - 2003 file format, you may encounter problems when trying to print or preview a [report](#) that has some margins set to 0. When you convert a version 2.0 report, margins are not set to 0; they are instead set to the minimum margin that is valid for the default printer. This is to prevent the report from printing data in the nonprinting region for the printer.

To resolve this problem, reduce the column width, column spacing, or number of columns in the report, so the width of the columns plus the width of the default margins is less than the width of your paper.

[My 16-bit API calls do not convert.](#)

If a Microsoft Access version or 2.0 module contains 16-bit API calls, you may

receive an error message when you convert the database to Access 2000 or Access 2002 2003 file format. You must change the API **Declare** statements in the converted database to their 32-bit equivalents.

The following list provides some tips for converting your code.

- Check any **Declare** statements to ensure that they refer to the correct dynamic-link libraries (DLLs). The following table shows the new names for the 32-bit Windows DLLs.

16-bit Windows DLL 32-bit Windows DLL

User.dll	User32.dll
Kernel.dll	Kernel32.dll
GDI.dll	GDI32.dll

- The names of some functions in the 32-bit Windows API have changed. Additionally, functions in the 32-bit Windows API are case-sensitive. Check to make sure that you've entered the procedure name and alias name correctly.
- Some functions have new parameter data types in the 32-bit Windows API.
- If a 16-bit version of a DLL with the same name as a 32-bit DLL exists on your computer, Access may try to call a function in that DLL if the directory in which it resides precedes the directory that contains the new version in your path.
- Some 32-bit DLLs contain functions with slightly different versions to accommodate both [Unicode](#) and [ANSI](#) strings. An **A** at the end of the function name specifies the ANSI version. A **W** at the end of the function name specifies the Unicode version. If the function takes string-type arguments, try appending an **A** to the function name.
- If your database application calls procedures in other DLLs, you must create or obtain 32-bit versions of those DLLs and make any necessary modifications to your code when you convert your database to the current version of Access. If you can't obtain a 32-bit version of a DLL, then you'll need an intermediary DLL that can convert 32-bit calls to 16-bit calls.

[My custom controls do not convert.](#)

When you convert a [Microsoft Access database](#) containing an [ActiveX control](#) to Access 2000 or Access 2002 - 2003 file format, you may receive an error.

Access version 2.0 supports 16-bit ActiveX controls, whereas Access 95 or later supports 32-bit ActiveX controls. Access can automatically update a 16-bit control to its 32-bit version only if a 32-bit version exists and is registered on your computer.

[I receive an error that a table exceeds the limit of 32 indexes.](#)

If your Microsoft Access version 2.0 database contains a table with 32 or more indexes and relationships combined, you may receive an error when you convert to Access 2000 or Access 2002 -2003 file format. To resolve this error, open your database in Access version 2.0 and modify your table design to reduce the number of relationships for the primary key table, or remove some indexes from the foreign key table. Then try to convert again.

[I receive an ODBC error when opening a form or report in a converted database.](#)

The message "ODBC – connection to *datasourcename* failed" occurs when a table in your converted Microsoft Access version 1.x or 2.0 database is linked to an [ODBC data source](#) that uses a 16-bit driver manager and driver. Microsoft Access 2002 or later can open only linked ODBC data sources that use the 32-bit versions of the ODBC Driver Manager (Odbc32.dll) and the appropriate ODBC driver— for example, the 32-bit version of Microsoft SQL Server ODBC Driver (Sqlsrv32.dll).

To correct this problem, you must create a new, identically-named 32-bit data source name (DSN) for each ODBC data source that is linked to the original Access 1.x or 2.0 database.

1. In Microsoft Windows Control Panel, double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**.
2. Create the new DSN by entering values identical to the original DSN. For example, for Microsoft SQL Server, you need to define at least **Data Source Name** and **Server**, but you may also need to define additional values such as the **Database Name**.

If you are uncertain what the original name of the DSN was, open the original database in the version of Microsoft Access that it was created with, open the linked table in [Design view](#), and then display the **Table Properties** sheet. The

table's **Description** property contains the definition for the ODBC connection string. The parameter following **DSN=** in the connection string is the name of the DSN.

[Troubleshoot converting an Access 2000 or later database to Access 97](#)

[I receive a message that my computer is missing at least one of the Access 97 object libraries.](#)

You can convert code that uses [Data Access Objects \(DAO\)](#) back to Microsoft Access 97, but you may receive a message that your computer is missing at least one of the Access 97 object libraries. Use this procedure to fix the missing references.

1. Convert the Access 2000 or later database to Microsoft Access 97. When you receive a message that your computer is missing at least one of the Access 97 object libraries, click **OK**.
2. Open the converted Access database in Microsoft Access 97.
3. Open a [module](#) in the database.
4. On the **Tools** menu, click **References**.
5. In the **Available References** box, clear the check boxes next to any missing references.
6. To set the correct reference, select the check box next to the Microsoft DAO 3.51 Object Library.

Note Unlike Access 2000 or later, Access 97 is not designed to work with [ActiveX Data Objects \(ADO\)](#).

[An Access 97 file that I converted from a security-enabled Access database in Access 2000 or Access 2002 - 2003 file format does not maintain its security measures.](#)

When you convert a Microsoft Access database in Access 2000 or Access 2002 - 2003 file format back to Access 97, permissions in the new Access 97 database are reset to their defaults because Access 97 can't use a workgroup information file in Access 2000 or Access 2002 - 2003 file format. You must help protect the new database in Access 97.

[Import specifications are missing from the Access 97 file I converted from Access 2000 or Access 2002 - 2003 file format.](#)

Import specifications created and saved in a Microsoft Access file in Access2000 or Access 2002 - 2003 file format are not created in the Access 97 file. You must recreate the specifications.

[Show All](#)

Troubleshoot undefined functions and references to projects or libraries

- When you open a Microsoft Access file that contains references to projects or libraries that aren't in the location specified in the **References** dialog box, Access tries to find the projects or libraries. However, if Access can't find a referenced file, you might have to set the reference manually. You might not be aware that you need to fix a missing reference until you receive an "undefined function" message or a message that Access can't find a project or library when you try to run a query or a Microsoft Visual Basic [procedure](#). To fix a missing reference, open a [module](#) in [Design view](#), and click **References** on the **Tools** menu. Then clear the check box next to any missing references and set the references you want.
- You can't call a class module from a query, form, report or macro. Store a Visual Basic procedure in a standard module if you need to call it from a query, form, report, or macro.
- You might receive an "undefined function" message when you open a [data access page](#) outside of Access if the data access page contains a function that isn't available outside of Access.

[Functions you can use in a data access page, or in PivotTable view or PivotChart view of a form](#)

A [control](#) in a [data access page](#) or a form that is open in [PivotTable view](#) or [PivotChart view](#) can contain an [expression](#) that uses a Microsoft Visual Basic for Applications (VBA) function. However, the expression will generate an error unless it appears on the following list.

Note This limitation doesn't apply to expressions in tables, queries, views, stored procedures, or user-defined functions that you have opened in PivotTable or PivotChart view.

[A - C](#)

ABS ARRAY

ASC ASCB
ASCW ATN
CBOOL CBYTE
CCUR CDATE
CDBL CHOOSE
CHR CHR\$
CHRB CHR\$
CHRW CHRW\$
CINT CLNG
COS CSNG
CSTR CVAR
CVDATE CVERR

[D - H](#)

DATE DATE\$
DATEADD DATEDIFF
DATEPART DATESERIAL
DATEVALUE DAY
DDDB ERROR
ERROR\$ EXP
FIX FORMAT
FORMAT\$ FV
HEX HEX\$
HOUR

[I - L](#)

IIF IMESTATUS
INSTR INT
IPMT IRR
ISDATE ISEMPY
ISERROR ISNULL
ISNUMERIC ISOBJECT

ICASE	ICASE\$
LEFT	LEFT\$
LEFTB	LEFTB\$
LEN	LENB
LOG	LTRIM
LTRIM\$	

[M - Q](#)

MID	MID\$
MIDB	MIDB\$
MINUTE	MIRR
MONTH	NOW
NPER	NPV
OCT	OCT\$
PARTITION	PMT
PPMT	PV
QBCOLOR	

[R - S](#)

RATE	RGB
RIGHT	RIGHT\$
RIGHTB	RIGHTB\$
RND	ROUND
RTRIM	RTRIM\$
SECOND	SGN
SIN	SLN
SPACE	SPACE\$
SQR	STR
STR\$	STRCOMP
STRCONV	STRING
STRING\$	SWITCH
SYD	

[T - Z](#)

TAN	TIME
TIMES\$	TIMER
TIMESERIAL	TIMEVALUE
TRIM	TRIM\$
TYPENAME	UCASE
UCASE\$	VAL
VARTYPE	WEEKDAY
YEAR	

[Show All](#)

Fix a reference to a library

In many cases, Microsoft Access will fix missing references automatically when you open an [Access file](#). Use the following procedure if you need to fix a missing reference manually.

1. Open a [module](#) or create a new module.
2. On the **Tools** menu, click **References**.
3. In the **Available References** box, clear the check boxes next to any missing references, and select the check box next to the reference or references you want.

[Show All](#)

About analyzing data

Microsoft Access provides several techniques to help you analyze data.

 [Create a crosstab query](#)

You can use a crosstab [query](#) to calculate and restructure data for easier analysis. Crosstab queries calculate a sum, average, count, or other type of total for data that is grouped by two types of information— one down the left side of the [datasheet](#) and another across the top.

1

Last Name	Category Name	Sum of Subtotal
Buchanan	Beverages	\$46,302.09
Buchanan	Condiments	\$16,789.35
Buchanan	Confections	\$36,182.13
Callahan	Beverages	\$111,047.76
Callahan	Condiments	\$49,566.21
Callahan	Confections	\$80,005.35

2

Last Name	Beverages	Condiments	Confections
Buchanan	\$46,302.09	\$16,789.35	\$36,182.13
Callahan	\$111,047.76	\$49,566.21	\$80,005.35

- 1 A select query
- 2 A crosstab query based on the select query

 [Add a PivotTable list to a data access page](#)

A PivotTable list is an interactive table that you can use to analyze data dynamically from within a Web browser. At run time, users can change the layout of the PivotTable list. They can move the fields that are displayed in the row, column, and data area of the PivotTable list, or add or remove fields from the list. Users can also filter, sort, and group data.

The following illustration shows a PivotTable list on the Orders data access page.

Orders	
ShipCountry ▾	
All	
	LastName ▾
	Davolio
	+ -
ShipName ▾	Sales ▾
Alfreds Futter	\$845.80
	\$471.20

[View a datasheet or form in PivotTable view](#)

Access 2002 or later provides a PivotTable view for [datasheets](#) and forms. In this view, you can build an interactive report using the fields underlying the datasheet or form. The view uses a Microsoft Office PivotTable Component, so you can do all the things that you can do on a PivotTable list. You can add fields to the view, move or remove fields, and filter, sort, and group data.

The following illustration shows the Employees table in PivotTable view.

Country ▾		
USA		
	Title ▾	
	Sales Represe	Grand T
	+ -	+ -
City ▾	First Name ▾	No Tot
Kirkland	Janet	
Redmon	Margaret	
Seattle	▶ Nancy	
Tacoma		
Grand To		

[Show All](#)

Add or edit data

[Add or edit data in a datasheet or form](#)

1. Open a [datasheet](#), or open a form in [Form view](#).
2. Do one of the following:

To add a new record, click **New Record**  on the toolbar, type the data, and then press TAB to go to the next field. At the end of the record, press TAB to go to the next record.

To edit data within a field, click in the field you want to edit, and then type the data.

To replace the entire value, move the pointer to the leftmost part of the field until it changes into the plus pointer , and then click. Type the data.

Notes

- To correct a typing mistake, press BACKSPACE. To cancel your changes in the current field, press ESC. To cancel your changes in the entire record, press ESC again before you move out of the field.
- When you move to another record, Microsoft Access saves your changes.

[Add new data in a data access page](#)

1. On the record navigation toolbar, click **New** .

Note If you don't see the **New** button, it means one of the following:

- The page does not support data entry.
 - The designer of the page has customized the button. Look for custom instructions on how to use the page, or contact the designer.
2. Type the data you want in the first field, and then press TAB to go to the next field.
 3. When you are finished adding data to each field, save the record.

Note If you make a typing mistake, press BACKSPACE. To cancel your changes in the current field, press ESC. To cancel your changes to the entire record, click **Undo**  on the record navigation toolbar.

[Edit data in a data access page](#)

1. Click in the field you want to edit.

Note You cannot edit group fields and [calculated fields](#). If you can't insert the input cursor inside any of the fields in the page, the page does not support edits.

2. Type the text you want to insert.
3. Save the record.

Note If you make a typing mistake, press BACKSPACE. To cancel your changes in the current field, press ESC. To cancel your changes to the entire record, click **Undo**  on the record navigation toolbar.

[Show All](#)

Save a record

[Save a record in a datasheet or form](#)

Microsoft Access automatically saves the record you are adding or editing as soon as you move the insertion point to a different record, or close the form or [datasheet](#) you are working on.

- To explicitly save the data in a record while you are editing it, click **Save Record** on the **Records** menu.

[Save a record in a data access page](#)

- On the record navigation toolbar, click **Save** .

Note If you don't see the **Save** button or if it appears disabled, it means one of the following:

- The designer of the page has customized the button. Look for custom instructions on the page on how to save your changes.
- The page is read-only and does not support additions or edits.

[Show All](#)

Delete a record

[Delete a record in a datasheet or form](#)

1. Open a [datasheet](#), or open a form in [Form view](#).
2. Click the record you want to delete.
3. Click **Delete Record**  on the toolbar.

Note When you delete data, you might want to delete related data in other tables. For example, if you delete a supplier, you probably want to delete the products that the supplier supplies. In some cases, you can make sure the proper data is deleted by enforcing [referential integrity](#) and turning on cascade deletions.

[Delete a record in a data access page](#)

Important Once you delete a record, you cannot undo the deletion from the database.

- On the record navigation toolbar, click **Delete** .

Note If you don't see the **Delete** button, it means one of the following:

- The page does not support deletions.
- The designer of the page has customized the button. Look for custom instructions on how to use the page, or contact the designer.

[Show All](#)

Undo changes

[Undo changes in Design view](#)

1. Click the arrow next to **Undo** .

Microsoft Access displays a list of the 20 most recent actions you can undo.

2. Click the action you want to undo.

When you undo an action, you also undo all actions above it in the list.

Notes

- If you later decide you didn't want to undo an action, click **Redo** .
- In a [Microsoft Access project](#), you can't undo changes in table [Design view](#) or in a database diagram.
- Access clears the Undo list when you switch to another view from Design view of a [data access page](#), a table in an Access database, or to a view, stored procedure, or function in an Access project. If you are working in Design view of a form that contains a subform and you switch to [Form](#) or [Datasheet view](#) or open the subform in its own window, Access removes from the list any recent actions that apply to the subform. Also, if you are working in Design view of a report that contains a subreport and you switch to Print Preview, Access removes from the list any recent actions that apply to the subreport.

[Undo changes when adding or editing records in a datasheet or form](#)

- Do one of the following:
- Click **Undo**  on the **Table Datasheet**, **Query Datasheet**, or **Form View** toolbar to take back your most recent change.
- Click **Undo Saved Record** on the **Edit** menu if you have already saved changes to the current record, or have moved to another record.

As soon as you begin editing another record, apply or remove a filter, or switch

to another window, you can't use these methods for correcting changes.

[Undo changes when adding or editing records in a data access page](#)

- Click **Undo**  on the record navigation toolbar.

Note If you don't see the **Undo** button or if it appears disabled, it means one of the following:

- The designer of the page has customized the button. Look for custom instructions on the page on how to undo your changes.
- The page is read-only and does not support additions or edits.

[Revert to a saved version of a form or report](#)

You can discard any changes—including changes to Microsoft Visual Basic code—that you have made in [Design view](#) of a form or report and revert to the last saved version of the form or report.

Caution Once you have reverted to the last saved version of the form or report, you cannot recover the changes that you have made to the form or report since you last opened it.

- To revert to the last saved version, on the **File** menu, click **Revert**.

Note This command is not available unless you have opened a form or report in Design view and have made changes to it. The **Revert** command is not available in Design view of a [data access page](#).

[Show All](#)

About compressing the data in a Text, Memo, or Hyperlink field (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Microsoft Access 2000 or later uses the [Unicode](#) character-encoding scheme to represent the data in a [Text](#), [Memo](#), or [Hyperlink](#) field. Unicode represents each character as two bytes, so the data in a Text, Memo, or Hyperlink field requires more storage space than it did in Access 97 or earlier, where each character is represented as one byte.

To offset this effect of Unicode character representation and to ensure optimal performance, the default value of the **Unicode Compression** property for a Text, Memo, or Hyperlink field is **Yes**. When a field's **Unicode Compression** property is set to **Yes**, any character whose first byte is 0 is compressed when it is stored and uncompressed when it is retrieved. Because the first byte of a Latin character— a character of a Western European language such as English, Spanish, or German— is 0, Unicode character representation does not affect how much storage space is required for compressed data that consists entirely of Latin characters.

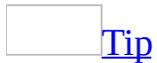
In a single field, you can store any combination of characters that Unicode supports. However, if the first byte of a particular character is not 0, that character is not compressed.

Data in a Memo field is not compressed unless it requires 4,096 bytes or less of storage space after compression. As a result, the contents of a Memo field might be compressed in one record, but might not be compressed in another record.

[Show All](#)

Expand a field, property box or text box to make it easier to edit

1. Open a [datasheet](#) or property sheet, or a form in [Form view](#).
2. Click the field you want to edit.
3. Press SHIFT+F2.



To expand a property box in a property sheet, you can also right-click the property box, and then click **Zoom**.

[Show All](#)

Insert a default value in a field

1. Open a [datasheet](#), or open a form in [Form view](#).
2. Click in the field in which you want to insert the default value.
3. Press CTRL+ALT+SPACEBAR.

Note To see the default value in a [Microsoft Access database](#), open the table in table [Design view](#) and look at the **DefaultValue** property. To see the default value in a [Microsoft Access project](#), display the Column Properties view in a [database diagram](#) and look at the **Default Value** column property.

[Show All](#)

Repeat the value from the previous record

In a field in a record, you can insert the value that is in the same field in the previous record.

1. Open a [datasheet](#), or open a form in [Form view](#).
2. Click in the field in which you want to insert the value.
3. Press CTRL+APOSTROPHE (').

[Show All](#)

Indicate nonexistent data with a zero-length string

To indicate nonexistent data in a Microsoft Access database, you can enter a [zero-length string](#) into a Text or Memo field. In a [Microsoft Access project](#) that is connected to a Microsoft SQL Server database, you can enter a zero-length string in a field whose [data type](#) is varchar or nvarchar.

1. Open a [datasheet](#), or open a form in [Form view](#).
2. To enter a zero-length string in the field, type two double quotation marks without a space in between (" ").

The field remains a zero-length string, even though the quotation marks disappear when you move to another field or object.

[Show All](#)

Turn on or off automatic correction of text and data

Use AutoCorrect to automatically correct text that you frequently mistype, and to automatically replace abbreviations with long names. For example, you can create an entry "esq" for "Martin R. Esquivel." Whenever you type **esq** followed by a space or punctuation mark, Microsoft Access automatically replaces "esq" with the name Martin R. Esquivel. AutoCorrect works with text in a [datasheet](#) and with text in a form in [Form view](#).



[Specify AutoCorrect rules, and exceptions to the rules](#)

1. On the **Tools** menu, click **AutoCorrect Options**.
2. Do one or more of the following:
 - To set capitalization options, select or clear the second through the fifth check boxes in the dialog box.
 - To correct your text by replacing with another string, select the **Replace text as you type** check box.

Note AutoCorrect won't correct text that was typed before you selected the **Replace text as you type** check box.

To add an entry that you want corrected automatically, do the following:

1. Select the **Replace text as you type** check box.
 2. In the **Replace** box, type the text that you want corrected automatically.
 3. In the **With** box, type the corrected text.
 4. Click **Add**.
- To specify exceptions to capitalization options, click **Exceptions**, and specify strings that you do not want corrected.

[Override AutoCorrect settings in Form view or Datasheet view](#)

When AutoCorrect replaces a string based on the rules specified in the **AutoCorrect Options** dialog box, it displays the AutoCorrect Options smart tag to the right of the corrected text. Click the smart tag, and do one of the following:

- To keep what you typed, click **Change...** or **Undo...** .

Note To go back to the corrected text, click the smart tag again, and click **Redo AutoCorrect**.

- To reverse the correction and update the list of rules or exceptions in the **AutoCorrect Options** dialog box, click **Stop...** .
- To display the **AutoCorrect Options** dialog box, click **Control AutoCorrect Options**.

Note You can control the display of the AutoCorrect smart tag by selecting or clearing the **Show AutoCorrect Options button** checkbox in the **AutoCorrect Options** dialog box.

[Add an AutoCorrect entry while checking spelling](#)

1. In the **Spelling** dialog box, select the correct item in the **Suggestions** box, or correct the text in the **Not In Dictionary** box.
2. Click **AutoCorrect** in the **Spelling** dialog box. When you receive a message asking if you want to use this word and continue checking, click **Yes**.

The next time you type the text as it appeared before you corrected it in the **Not in Dictionary** box, Microsoft Access automatically changes it to the corrected text.

Note By default, AutoCorrect is enabled for all text boxes and combo boxes in Form view. To disable this feature for a specific control, set the control's **AllowAutoCorrect** property to **No**.

I can't display my records again after using the Data Entry command. (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

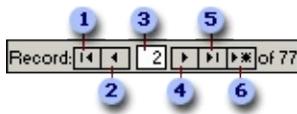
- On the **Records** menu, click **Remove Filter/Sort**.

[Show All](#)

Move between records or fields

[Move between records by using navigation buttons in a datasheet or form](#)

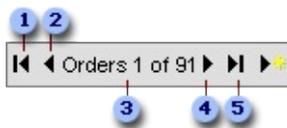
The navigation buttons are located at the bottom of the window in [Datasheet](#) or [Form view](#). You can use these buttons to move quickly between records.



- 1 First record
- 2 Previous record
- 3 Record number
- 4 Next record
- 5 Last record
- 6 New record

[Move between records by using navigation buttons on a data access page](#)

- Use the navigation buttons located on the record navigation toolbar to quickly move between records or a group of records on a [grouped data access page](#).



- 1 First record
- 2 Previous record
- 3 Record source and number

4 Next record

5 Last record

Note If you don't see the record navigation toolbar on your page, it means one of the following:

- All the records are displayed on the page, so the navigation buttons are not necessary.
- The designer of the page has customized the toolbar. Look for custom instructions on how to use the page, or contact the designer.



[Move between fields with the Go To Field box in Datasheet view](#)

Adding the **Go To Field** box makes it easier to move between fields in long records in [Datasheet view](#).

1. Open a table, query, form, view, or stored procedure in Datasheet view.
2. Click **Toolbars** on the **View** menu.
3. Select **Formatting (Datasheet)**.
4. Click **Close**.

The **Go To Field** box is the leftmost item on the **Formatting (Datasheet)** toolbar. Use it to click the name of the field that you want to move to in the current record.

[Show All](#)

Record selector symbols

To show the status of the current record in a form or datasheet, Microsoft Access displays the following symbols in the current [record selector](#).

Symbol	Meaning
	This is the current record; the record has been saved as it appears.
	You are editing this record; changes to the record aren't yet saved.
	You are not currently editing this record in a batch transaction; changes to this record aren't yet saved. (Microsoft Access project only).
	This record is locked by another user; you can't edit it (Microsoft Access database only).
	This is a new record that you can enter information in.
	This field is the primary key field and contains unique data to identify the record (table Design view only).

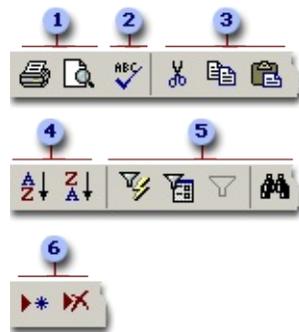
[Show All](#)

About working with data in a datasheet

In a table or query, [Datasheet view](#) provides the tools you need to work with data.

[Using the Table Datasheet and Query Datasheet toolbars](#)

The **Table Datasheet** and **Query Datasheet toolbars** provide many of the tools you need to find, edit, and print records.



- 1 Print or preview data.
- 2 Check spelling.
- 3 Cut, copy, or paste selected text, fields, whole records, or the entire datasheet.
- 4 Sort records.
- 5 Filter records, and find or replace values.
- 6 Add or delete records.

[Working with columns, rows, and subdatasheets](#)

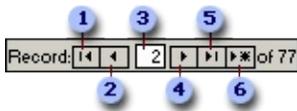
You can find tools for working with columns, rows, and [subdatasheets](#) in the datasheet itself, or by right-clicking a [column selector](#).



- 1 Use the column selector to move, hide, or rename a column.
- 2 Resize columns or rows.
- 3 Use subdatasheets to view related data.
- 4 Freeze the leftmost column so that it is displayed as you scroll to the right.

[Moving through records](#)

You can use the navigation toolbar to move through the records in a datasheet.



- 1 Go to the first record.
- 2 Go to the previous record.
- 3 Type a record number to move to.
- 4 Go to the next record.
- 5 Go to the last record.
- 6 Go to a blank (new) record.

[Show All](#)

Troubleshoot editing data in a field in Datasheet or Form view

[Nothing happens when I try to edit data.](#)

There are several kinds of fields in forms or datasheets that you can't edit. The following fields display values just as regular fields do, but if you try to enter data into them, nothing happens:

- **AutoNumber fields** If a field in a [Microsoft Access database](#) has an [AutoNumber data type](#) or in a [Microsoft Access project](#) has a numeric data type with the **Identity** property set, Access automatically assigns a number to this field for each record you add. This type of field is often used as a record ID number or [primary key](#). When you start adding a new record, Access automatically fills in the field's value for you and you can't edit it.
- **Calculated fields** In an Access database, a [calculated field](#) displays values that Access calculates. They are usually based on other fields in your tables, but calculated field values are not stored in tables.
- **Locked or disabled fields** If a control on a form has the **Locked** property set to **Yes** or the **Enabled** property set to **No**, you can't edit the data in the field.
- **Fields in a locked record** If you use an Access database in a multiuser environment and a record is locked by another user, you can't edit the data in the record. When you move to a locked record, the locked record indicator  is displayed in the record selector.
- **Fields in snapshots** While most queries return recordsets that you can edit, some queries return [snapshots](#), which can't be edited.
- **Fields in a read-only form or a locked database** If the form you're using has the **AllowEdits** property set to **No** or if the underlying data is read-only or locked by another person, you can't edit data in the form.

[After I've added or edited data, I receive a message and I can't move to another field or record.](#)

When you try to move out of a field after changing data, Microsoft Access

validates the data, making sure that the value you entered is allowed in that field. If the value isn't allowed, Microsoft Access alerts you. In order to move off the field, you must either change the data to an acceptable value or undo your changes. If the record can't be saved, you must either make any necessary corrections or click **Undo**  to cancel all changes you made to the record.

A value might not be allowed in a field for the following reasons:

- The value isn't compatible with the field's [data type](#). If you enter a value that doesn't match the data type, Microsoft Access attempts to convert the entry to the correct data type. However, some data can't be converted to other data types. For example, you can't store a name in a field with the Number data type in a [Microsoft Access database](#) or the smallint data type in a [Microsoft Access project](#).
- You may be having problems entering dates in a field with the Date/Time data type. If you are entering dates using the slash mark (/) format, don't enter a trailing slash mark. For example, entering 1/3 is valid, but 1/3/ is not valid.
- The data breaks a rule set in the **ValidationRule** property for the field.
- Data is required in the field, and you've tried to leave it blank.
- The validation macro specified for the **BeforeUpdate** property canceled the update.
- The field does not accept [zero-length strings](#). In an Access database, you can only enter a zero-length string in a Text or Memo field whose **AllowZeroLength** field property is set to **Yes**. In an Access project, you can only enter a zero-length string in a field with the varchar or nvarchar data type in a Microsoft SQL Server 7.x database. Zero-length strings are not supported in SQL Server 6.5.
- You tried to enter a duplicate value in a field that is a [primary key](#) or that has a [unique index](#).

[Show All](#)

Select fields and records

[Select fields and records in a datasheet](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .
Multiple records	The record selector of the first record, and then drag to extend the selection.
All records	Select All Records on the Edit menu.

[Select fields and records in a data access page](#)

The following table lists mouse techniques for selecting data or records in [Page view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data
An entire field	Three times within the field
A record	A field that belongs to the record

The following table lists keyboard techniques for selecting data or records in Page view.

To select	Do this
-----------	---------

Data in a field Move the insertion point to the start of the text you want to select, and then hold down SHIFT and press the arrow keys to move to the end of the selection.

An entire field Move the insertion point to anywhere in the field, press HOME, and then press SHIFT+END.

A record Press TAB until a field that belongs to the record gets focus.

Note Your data access page might have a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component, which have their own methods of selection. To display Help for one of these components, click **Help**  on its toolbar.

[Show All](#)

Copy or move data

If you are pasting data from multiple fields to a [datasheet](#), make sure the columns match the order of the data you want to copy or move.

When you paste records or data from multiple fields to a form, Microsoft Access pastes the data into fields with the same name as the source fields, regardless of their order on the form. If the data you want to copy or move contains fields that don't exist in the destination form, Access asks you if you want to paste only fields with matching names. If there are no matching field names, Access pastes the fields according to the destination form's tab order—this might not be the order you want. If the field names in the source are different from the field names in the destination, you might want to paste the data into a datasheet instead of a form.

[Copy or move data from one field to another](#)

1. Open a datasheet, or open a form in [Form view](#).
2. Select the data you want to copy or move.

[How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.

A record The [record selector](#).
Multiple records The record selector of the first record, and then drag to extend the selection.
All records **Select All Records** on the **Edit** menu.

3. To copy the data, click **Copy**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

To move the data, click **Cut**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

4. To replace the current value of the destination field, select the whole field.

To insert the data into the existing data, position the insertion point where you want to paste the data.

5. Click **Paste**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

[Copy or move records from another application to Microsoft Access](#)

Be sure the data is arranged in a spreadsheet or table, or separated by tab characters, before you select it.

1. In the other application, select one or more rows you want to copy or move.
2. Copy or cut the records in the other application.
3. In Microsoft Access, open the datasheet or form in which you want to paste the records.
4. Do one of the following:
 - To replace records, select those records, and then click **Paste**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar. If the [record selector](#) isn't visible in a form, click **Select Record** on the **Edit** menu.
 - To append the data as new records, click **Paste Append** on the **Edit** menu.

[Copy or move records or data from multiple fields in Microsoft Access to another application](#)

When you paste records into another application, Microsoft Access pastes the field names in the first row, followed by the data.

1. Open a datasheet, or open a form in [Form view](#).
2. Select the data you want to copy or move.

[How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .
Multiple records	The record selector of the first record, and then drag to extend the selection.
All records	Select All Records on the Edit menu.

3. To copy data, click **Copy**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

To move data, click **Cut**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

4. Open the application in which you want to paste the data.
5. Do one of the following:
 - To replace data with the data from Access, select the data you want to replace.
 - To insert the data into existing data, position the insertion point where

you want to paste the data.

- To paste into a Microsoft Excel spreadsheet, select the cell in which you want the first column heading.
- To paste into a Microsoft Word for Windows document, move the insertion point to where you want to place the records. The records are pasted as a table in Word.

6. Paste the data.

Note If you are copying from a datasheet that contains subdatasheets, Access can copy only one level of datasheet at a time.

[Copy or move records or data from multiple fields to a datasheet](#)

Before copying or moving the data, make sure the columns in the destination datasheet match the order of the data you want to copy or move.

1. Open a datasheet, or open a form in [Form view](#).
2. Select the data you want to copy or move.

[How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .

Multiple records The record selector of the first record, and then drag to extend the selection.

All records **Select All Records** on the **Edit** menu.

3. To copy data, click **Copy**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

To move data, click **Cut**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

4. Open the datasheet you want to paste the data into. If necessary, rearrange the columns of the datasheet to match the order of the data you are copying or moving.
5. To replace the current data in the destination, do one of the following:
 - If you are pasting data from multiple fields, select a destination that has the same number of fields and records as the data you copied, and then click **Paste**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.
 - If you are pasting records, select the records you want to replace, and then click **Paste**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

To paste the data as new records at the end of the datasheet, click **Paste Append** on the **Edit** menu.

Note If the source datasheet has more fields than the destination datasheet, Microsoft Access doesn't paste the extra fields.

 [Copy or move records or data from multiple fields to a form](#)

1. Open a datasheet, or open a form in [Form view](#).
2. Select the data you want to copy or move.

 [How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select

Click

Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .
Multiple records	The record selector of the first record, and then drag to extend the selection.
All records	Select All Records on the Edit menu.

3. To copy data, click **Copy**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

To move data, click **Cut**  on the **Table Datasheet, Query Datasheet, or Form View** toolbar.

4. Open the form you want to paste the data into.
5. To replace the current data in the destination form, select the data you want to replace, and then click **Paste**  on the **Database** toolbar.

To paste the data as new records, click **Paste Append** on the **Edit** menu.

[Show All](#)

Copy or move a control or other item

You can copy or move items such as fields, [controls](#), text, or macro [actions](#). When you copy a field, control, or macro action, Microsoft Access includes any associated properties, controls, or [action arguments](#). For example, when you copy a text box control, Access copies its label as well. When you copy a macro action by clicking the [row selector](#), Access copies the associated action arguments, macros, and [conditional expressions](#). However, Access doesn't copy [event procedures](#) associated with a control.

When you select an item and then use the **Copy** or **Cut** command, Microsoft Access replaces the contents of the Clipboard with that item. As long as the item is stored on the Clipboard, you can paste it wherever and whenever you want. Also, if you are in [Design view](#) of a form, report, or [data access page](#), you can use the **Duplicate** command to create a copy of a control without replacing the contents of the Clipboard.

[Copy or move a control or other item](#)

1. Select the items you want to copy or move.
2. Do one of the following:

To copy the item, click **Copy**  on the toolbar.

To move the item, click **Cut**  on the toolbar.

3. Click the location where you want to place the item.
4. Click **Paste**  on the toolbar.

Notes

You can use the Microsoft Office Clipboard to collect and paste multiple data items such as records or groups of records, or the contents of fields or controls. However, you cannot use the Office Clipboard to collect items such as controls, macro actions, or database objects.

To a limited extent, you can determine where Microsoft Access pastes a control.

If you select a section, Access pastes the control in the upper-left corner of the section. If you select a control near the area where you want to paste the control, Access pastes the control just below the selected control. Note, however, that if you paste a label on a form or report, and the control that you select doesn't have an attached label, Access attaches the label to the selected control.

[Duplicate a control](#)

1. In Design view of a form, report, or data access page, select the control you want to duplicate. If the control has an attached label, click the control, not the label.
2. On the **Edit** menu, click **Duplicate**.

Microsoft Access pastes the control just below the control you selected.

3. If you want to create another evenly spaced control, click **Duplicate** again.

Microsoft Access places the control beneath the most recently pasted control, even if you moved that control.

[Show All](#)

About pasting data from another application

When you copy a selection from another application to a form or report in Microsoft Access and click **Paste Special** on the **Edit** menu, Microsoft Access displays the **Paste Special** dialog box. The following table describes the options in the **As** list in the **Paste Special** dialog box.

Click this option in the As box	If you want to
Application name (for example, Microsoft Excel Worksheet)	Display the object in an unbound object frame . You can edit the object directly from the form or report if you click this entry.
Picture or Bitmap	Display the object in an image control . Click one of these entries if you want to optimize the performance of your form or report. You can't edit the object from the form or report if you click one of these entries.
Text	Display the text that you copy (for example, the text in a Microsoft Excel cell) in a label if the object is inserted , or a text box if the object is linked . When you link a text box this way, Microsoft Access creates a DDE link in the text box. When you switch to Form view or Print Preview, you'll see the actual linked text.

[Show All](#)

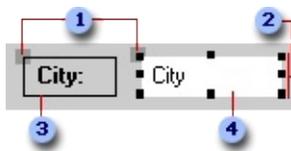
Delete a control or another item

1. Do one of the following:

Select the control or controls that you want to delete.

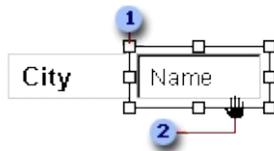


In a form or report, if the [control](#) has an attached [label](#), it looks like this:



- 1 Move handles
- 2 Sizing handles
- 3 Attached label
- 4 Selected control

In a [data access page](#), if the control has an attached label, it looks like this:



- 1 Move handles
- 2 When the pointer changes to an open hand, you can drag the control.
 1. Open a form, report, or data access page in Design view.
 2. Do one of the following:
 - To select one control, click anywhere in the control.
 - To select adjacent controls, click at any point outside a control,

and drag a rectangle over the controls that you want to select.

- To select nonadjacent or overlapping controls, hold down the SHIFT key and click each control that you want to select.

Note In a data access page, you can't select more than one control unless you have installed Microsoft Internet Explorer 5.5 or later.

3. If you want to remove one or more controls from the selected group, hold down the SHIFT key, and then click the controls that you want to remove.

Note If you click a control that's already selected, an insertion point appears inside the control so that you can enter or edit text. To reselect the control, click outside the control, and then click the control again.



You can specify whether the rectangle that you drag over controls needs to only touch the controls or must enclose them completely to select them. On the **Tools** menu, click **Options**, click the **Forms/Reports** tab, and then click one of the options under **Selection Behavior**.

Select the field, text, [macro](#) action, or other item that you want to delete.

2. Press DELETE.

Note If you are deleting a control that has an attached label, Microsoft Access deletes the control and its label. If you want to delete just the attached label, click the label and press DEL.

[Show All](#)

Troubleshoot problems when pasting data

 [Microsoft Access can't paste the data.](#)

In most cases, if Microsoft Access can't paste some or all of the data in a [Microsoft Access database](#), it creates the Paste Errors table. In that case, Microsoft Access displays a message telling you that data will be added to the Paste Errors table.

There are several reasons why Microsoft Access will not paste some or all of your data:

- You tried to paste an entire field or fields without selecting a destination. You must select the field or fields that you want to paste to.
- You tried to paste more fields than the datasheet you're pasting to has, or more than the number of fields you selected as a destination. Microsoft Access doesn't paste the extra fields.
- You tried to paste a value that's not compatible with the [data type](#) of the field. For example, you tried to paste letters into a field in a Microsoft Access database with the Number data type or in a [Microsoft Access project](#) with the smallint data type.
- You tried to paste text that's too long for the field. The setting in the **FieldSize** property of an Access database or the **Length** column property in an Access project determines the maximum size of a field value. Check the field in [table Design view](#) in an Access database or Column Properties view in a [database diagram](#) in an Access project to see the setting.
- You tried to paste a duplicate value into the table's [primary key](#), into a field that has the **Indexed** property set to **No Duplicates** in an Access database, or into a field with either a unique constraint or unique index in an Access project. Check the field in table Design view in an Access database or Column Properties view in a database diagram in an Access project to see the setting.
- You tried to paste data into a hidden field. This is not permitted. To display hidden columns, click **Unhide Columns** on the **Format** menu.

- In an Access database, you tried to remove data from or paste data into a [disabled](#), [locked](#), [calculated](#), or [AutoNumber field](#).
- You don't have [permissions](#) to add or edit data in the database object.
- A value you tried to paste isn't allowed in the field based on the settings of the **ValidationRule**, **InputMask**, **Required** (Access database only), or **AllowZeroLength** (Access database only) properties.
- The validation macro specified for the **BeforeUpdate** or **OnInsert** properties canceled the update. The data you tried to paste did not meet the validation requirements.

 [The data I pasted is in the wrong field.](#)

- When you paste records into a datasheet, Microsoft Access places them in the order they appear in the columns of the datasheet, regardless of the names of the fields. If you want the data to be pasted according to corresponding field names, paste them into a form in [Form view](#).
- When you paste records into a form, Microsoft Access places them into fields with the same name as the source fields, regardless of their order on the form. However, if there are no matching field names, the data is placed according to the form's tab order, which may not be the order you intend.

[Show All](#)

About finding or replacing data

There are many ways that Microsoft Access can find or replace just the data you need, whether you're looking for a particular value, one record, or a group of records.

- You can find a record by scrolling in a [datasheet](#) or form, or by typing its record number in the [record number box](#).
- With the **Find** dialog box, you can locate specific records or find certain values within fields. You can navigate through records as Access finds each occurrence of the item you're looking for. If you want to replace certain values that you find, use the **Replace** dialog box instead.
- With a [filter](#), you can temporarily isolate and view a specific set of records to work with while you have a form or datasheet displayed.
- With a [query](#), you can work with a specific set of records that meet criteria you specify from one or more tables in the database. You can work with this subset independent from a specific form or datasheet by running the query.
- In a [Microsoft Access database](#), you can replace large amounts of data more quickly or perform calculations on data (such as raising all salaries in an Employee table by 5 percent) by using an [update query](#) instead of the **Replace** dialog box. In a [Microsoft Access project](#), you can use a [stored procedure](#) to update large amounts of data. However, you can't confirm individual replacements when using an update query or a stored procedure. Also, an update query or stored procedure isn't as convenient when you want to search for and replace data in more than one field.

Note When you attempt to find or replace data in an Access project, Access performs the operation only on the data that is stored on the client. If your datasheet or form has reached its [maximum record limit](#), there may be more data on the server. To ensure that you are finding or replacing all the data, you might want to specify that there is no limit on the maximum number of records.

[Show All](#)

Find or replace a value in a field

When looking for or replacing values in a [datasheet](#) with a subdatasheet, or a form with a subform displayed as a datasheet, Microsoft Access only looks in the datasheet, subdatasheet, or form where you've placed the insertion point.

1. In [Form](#) or [Datasheet view](#), select the field (column) you want to search, unless you want to search all fields. (Searching a single field is faster than searching the entire datasheet or form.)
2. Do one of the following:

[Find specific occurrences of a value in a field](#)

1. Click **Find**  on the **Form View** or **Datasheet View** toolbar.
2. In the **Find What** box, type the value you want to find.

If you don't know the exact value you want to find, you can use wildcard characters in the **Find What** box to specify what you're looking for.

3. Set any other options you want to use in the **Find and Replace** dialog box. For more information about an option, click , and then click the option.
4. Click **Find Next**.

[Find blank fields or zero-length strings in an Access database](#)

1. Click **Find**  on the **Form View** or **Datasheet View** toolbar.
2. Do one of the following:
 - If blank fields are formatted to show a string (such as "Unknown"), type the formatted string in the **Find What** box, and make sure that the **Search Fields As Formatted** check box is selected.
 - To find unformatted blank fields, type **Null** or **Is Null** in the **Find What** box, and make sure that the **Search Fields As Formatted**

check box is not selected.

- To find [zero-length strings](#) in a [Microsoft Access database](#), in the **Find What** box, type double quotation marks (""), with no spaces in between, and make sure that the **Search Fields As Formatted** check box is not selected.
3. Set any other options you want to use in the **Find and Replace** dialog box. For more information about an option, click , and then click the option.
 4. To find one or more occurrences of a blank field, click **Find Next**.

[Replace specific occurrences of a value in a field](#)

1. On the **Edit** menu, click **Replace**.
2. In the **Find What** box, type the value you want to find; in the **Replace With** box, type the value you want to replace it with.

If you don't know the exact value you want to find, you can use wildcard characters in the **Find What** box to specify what you're looking for.

3. Set any other options you want to use in the **Find and Replace** dialog box. For more information about an option, click , and then click the option.
4. Do one of the following:
 - To replace all occurrences of the specified value at once, click **Replace All**.
 - To replace each occurrence one at a time, click **Find Next**, and then click **Replace**; to skip an occurrence and find the next one, click **Find Next**.

[Show All](#)

Find a record in a datasheet or form

 [Find a specific record by scrolling in a datasheet or form](#)

If there are more records than fit in your [Datasheet view](#) or [continuous form](#) window, you can use the scroll box on the scroll bar to find a specific record.

- Drag the scroll box up or down on the scroll bar.

Next to the scroll box, Microsoft Access displays the number of the record that is at the top of the screen.

 [Find a record by record number in a datasheet or form](#)

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#), or a form in [Form view](#).
2. Double-click the number in the [record number box](#) to select it, or press F5.
3. Type the number of the record you want, and then press ENTER.

[Show All](#)

Troubleshoot finding and replacing data

[I can't find or replace values that I know exist.](#)

- If the **Find and Replace** dialog box doesn't find records that meet the criteria you specified, and you know the records exist, check whether the field you're searching in is a [Lookup field](#). A Lookup field looks up values from another table and stores the references to values in the current table. If it is a Lookup field, you need to select the **Search Fields As Formatted** check box in order for Microsoft Access to find the records.
- When you try to find or replace data in a [Microsoft Access project](#), Access performs the operation only on the data that is stored on the client. If your [datasheet](#) or form has reached its [maximum record limit](#), there is more data on the server. To find or replace all the data on the server, set the maximum record limit to **no limit** by using the maximum record limit button to the right of the [navigation buttons](#).
- When looking for values in a datasheet with a subdatasheet, or form with a subform displayed as a datasheet, Microsoft Access only looks in the datasheet, subdatasheet, or form where you've placed the insertion point.
- When you use the **Find and Replace** dialog box in a form, Access doesn't search for values in [unbound controls](#).
- If Access doesn't find expected values when replacing date or time values, try selecting the **Search Fields As Formatted** check box.

[I can't find or replace text in a field with an input mask.](#)

If you add an [input mask](#) to a table that already contains data, the **Find and Replace** dialog box may not find the text when the field's data does not meet the criteria of the input mask. To resolve this, you can remove the input mask, use the **Find and Replace** dialog box, and then add the input mask back. Alternatively, you can modify the data in the field with the input mask to meet the criteria of the input mask.

[It takes too long to find and replace data.](#)

For the fastest searches, search for whole field values or the first character within a single [indexed](#) field. If you search the same nonindexed field repeatedly, index the field.

[I can't find the data I want when I use wildcards.](#)

In a [Microsoft Access database](#) where the [ANSI SQL query mode](#) has been set to ANSI-92 SQL, you must use ANSI-89 wildcards instead of ANSI-92 wildcards in the **Find and Replace** dialog box. For example, if you are searching for data that begins with the letter "B", type **B*** in the **Find What** box instead of **B%**.

[Show All](#)

About Filters (MDB)

 [Recommendations for choosing a filtering method in a table, query, or form](#)

[Filter By Selection](#), [Filter By Form](#), and [Filter For Input](#) are different ways to filter records in a form or datasheet.

- If you can easily find and select the value you want the filtered records to contain, use Filter By Selection.
- If you want to choose the values you're searching for from a list without scrolling through all the records, or if you want to specify multiple criteria at once, use Filter By Form.
- If the focus is in a field and you just want to type in place the exact value you're searching for or the [expression](#) whose result you want to use as your criteria, or if you want to specify multiple criteria at once, use Filter For Input.

For complex filters, use [Advanced Filter/Sort](#). You can:

- Search for records that meet multiple criteria.
- Search for records that meet one criteria or another criteria.
- Enter expressions as criteria.

Note After specifying the values you're searching for by using Filter By Selection, Filter By Form, or Filter For Input, you can switch to the Advanced Filter/Sort window to see how the values you specified are translated into a criteria expression in the design grid.

 [Similarities and differences between select queries and filters](#)

The basic similarity between [select queries](#) and [filters](#) is that they:

- Both retrieve a subset of records from an underlying table or query.
- Produce results that can be used as the source of data for a form or report.
- Can sort records.
- In general, enable you to edit data if editing is otherwise allowed. (You can also perform bulk updates with an [update query](#).)

How you want to use the records that are returned determines whether you use a filter or a query.

Generally, use a filter to temporarily view or edit a subset of records while you're in a form or datasheet. Use a query if you want to do any or all of the following:

- View the subset of records without first opening a specific table or form.
- Choose the tables containing the records you want to work with and add more tables at a later date if necessary.
- Control which fields from the subset of records display in the results.
- Perform calculations on values in fields.

Note Even if you determine you need a query, consider taking advantage of the easy [Filter By Form](#), [Filter By Selection](#), or [Filter For Input](#) techniques for creating a filter, and then saving the filter as a query. This enables you to bypass the query's [design grid](#) altogether (unless you want to make additional changes to the query). Even so, when you open the query in [Design view](#), you'll see how Microsoft Access filled in the design grid using the information from the filter. You can then use this information as a guide for making additional changes.

[How filters are saved for reuse](#)

The way filters are saved after they are created or applied varies depending on the object.

- When you save a table or form, Microsoft Access saves the filters you created. You can reapply the filters when you need them, the next time you open the table or form.
- When you save a query, Access saves the filters you created, but it does not add the filter criteria to the query design grid. You can reapply the filters after you run the query, the next time you open it.
- If you created a filter on a subform or [subdatasheet](#), this filter is also available when you open the form or table for the subform or subdatasheet

independently.

- In a report, you can't create a filter by using the filtering techniques mentioned previously. However, a report or form can inherit a filter.

[How a filter and sort order for a table or query affect new forms or reports](#)

When you create a form or report based on a table or query that has a filter or a sort order specified in it, Microsoft Access includes the filter and sort order with the form or report. The sort order is automatically applied. Whether or not the filter is automatically applied depends on how you create the form or report.

- If you create the form or report directly from filtered data while the data is displayed in an open table or query, the filter is applied automatically each time you view the report, but it is applied only the first time you view the form after creating it. (The next time you open the form, you'll apply the filter yourself.)
- If you create the form or report based on a closed table or query, the filter won't be applied in the new form or report. You apply it when you need it.

[Ways to filter records for a data access page](#)

[Ways to filter records before they are retrieved from the database](#)

Filtering records before they are retrieved from the database can improve data access page performance because there are fewer records to download from the underlying database.

- You can base your page on a [query](#) that includes criteria (search conditions) to limit the records.
- In a [Microsoft Access database](#) you can define a server-side filter in the **ServerFilter** property of the **RecordsetDef** object, or by using the **Filter** property in Visual Basic for Applications (VBA) or Microsoft Visual Basic Scripting Edition (VBScript). In a [Microsoft Access project](#) you can define a server-side filter in the **ServerFilter** property in VBA or VBScript.
- On a grouped data access page, you can create a filter that retrieves data for a specific group, such as orders for a specific customer. The data access page won't display any records until you choose the group from a drop-

down list in [Page view](#) or Microsoft Internet Explorer. Once you choose the group, the filter is applied and only those records are retrieved.

On an ungrouped data access page, you can create a filter by designating a field as the filter field. When you choose a value from the filter field, all records that contain the selected value are retrieved.

- In [Design view](#), you can add a server-side filter to a hyperlink on a page. Then, when you click the link in Page view or in the browser, only the records specified in the filter are displayed on the page you jump to.

[Ways to filter records after they are retrieved from the database](#)

Use the [Filter By Selection](#) technique in [Page view](#) or the browser. With this type of filter, you select in a field the value that you want the filtered records to contain.

[About filtering data in a PivotTable or PivotChart view](#)

[About filtering data in a PivotTable view](#)

You can use the filter feature to find specific data values or all data that matches a value.

[Filtering techniques](#)

Filtering a field (Autofiltering) You can filter fields in the [row](#), [column](#), and [detail areas](#). When you filter a field, you select one or more [items](#) of data in the field that you want to view, and hide the other items.

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr4	\$2,000
Tennis	Qtr3	\$600
Tennis	Qtr4	\$1,500

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr4	\$2,000
Golf	Qtr3	\$6,430

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr3	\$6,430

- 1 First, the Sport field is filtered to display only Golf sales ...
- 2 ... and then the Quarter field is filtered to display only Golf sales in Qtr3.
- 3 When you filter a field, the drop-down arrow ▼ for the filtered field changes to blue instead of black, and the **AutoFilter** button  on the toolbar is selected.

Filtering a field is particularly helpful when you have a large amount of source data but you want to focus on specific areas. When you filter a field, you can display the data for a single item, or you can select some items to display and other items to hide. You can filter on more than one field at a time to further narrow the focus.

Filtering based on the data in one cell (Filter by Selection) You can filter a field to display only data that matches the value in a selected cell. Filtering by selection is particularly useful for fields in the detail area, when you want to view all of the rows that contain a particular value.

Using a field in the Filter axis You can also filter data by using [filter fields](#). When you select one or more items in the filter field, the data that's displayed and calculated in the entire PivotTable view changes to reflect those items. You can add new filter fields or move existing fields to the filter area.



1 Filter field

Conditional filtering You can apply a filter to a row or column field to show the top or bottom n items based on a total. For example, you can filter for the three cities that generated the most sales or the five products that are least profitable. Instead of n items, you can also choose to filter for a certain percentage of items. For example, you can filter for the top or bottom 25% of salespeople based on the number of orders handled. If there are 40 salespeople, after the filter is applied, you will see data for 10 people.

[Combining multiple filters](#)

- Filters are additive. Each filter you apply is added to the ones you already have in effect. However, when you apply a [filter by selection](#), the [conditional filter](#) already applied on the field is removed before the filter by selection is applied.
- Conditional filters are independent of each other. The result of one conditional filter does not affect the result of another.
- An [autofilter](#) is always applied before a conditional filter, regardless of the order in which you applied them. For example, if you set a conditional filter to show the top two cities based on sales, followed by an autofilter on the ShippedCity field to include only five cities, the PivotTable view will show the top two of the five cities you selected.

[Removing and reapplying a filter](#)

You can display all data for all fields by turning off [autofiltering](#). However, the filter settings are retained, so that when you turn autofiltering back on, the data that was previously displayed or hidden is again displayed or hidden.

Filter settings are retained when you move fields to change the layout. Also, if you remove a field and later add the field back to the layout, the same items are again hidden.

If you don't want to retain your filter selections, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Identifying filters that are currently in effect](#)

- When items in a field are hidden by [autofiltering](#) or [filter by selection](#), the drop-down arrow  in the field label is blue. Click the blue arrow  to see the filter that is in effect.
- When items in a field are hidden by [conditional filtering](#), a funnel  appears to the left of the drop-down arrow . You can get more information

about the current filter in the **Properties** dialog box.

[Effects of filtering on calculations](#)

Totals can include only the displayed data, or both visible and hidden data. The default is to exclude hidden data from totals. You can change the setting to include or exclude hidden data in totals. This setting affects only data that's hidden by filtering row and column fields; data that's hidden by using [filter fields](#) is always excluded from totals.

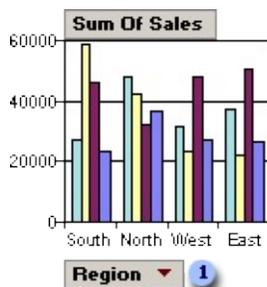
[About filtering data in a PivotChart view](#)

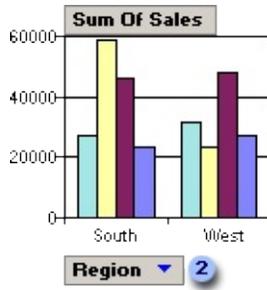
You can use the filter feature to find specific data values or all data that matches a value.

[Filtering techniques](#)

Filtering a field (Autofiltering) You can filter fields in the category, series, and filter areas. When you filter a field, you select one or more items of data in the field that you want to view, and hide the other items.

Filtering a field is particularly helpful when you have a large amount of source data but you want to focus on specific areas. When you filter a field, you can display the data for a single item, or you can select some items to display and other items to hide. You can filter on more than one field at a time to further narrow the focus.





- 1 Category field Region showing all items (not filtered)
- 2 Category field Region filtered to show South and West region items

Conditional filtering You can apply a filter to a [series](#) or [category field](#) to show the top or bottom n items based on a total. For example, you can filter for the three cities that generated the most sales or the five products that are least profitable. Instead of n items, you can also choose to filter for a certain percentage of items. For example, you can filter for the top 25% or bottom 25% of salespeople based on the number of orders handled. If there are 40 salespeople, after the filter is applied, you will see data for 10 people.

[Combining multiple filters](#)

Filters are additive. Each filter you apply is added to the ones you already have in effect.

[Removing and reapplying a filter](#)

You can display all data for all fields by turning off [autofiltering](#). However, the filter settings are retained, so that when you turn autofiltering back on, the same data that was previously displayed or hidden is again displayed or hidden.

Filter settings are retained when you move fields to change the layout. Also, if you remove a field and later add the field back to the layout, the same items are again hidden.

If you don't want to retain your filter selections, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Identifying the filters that are currently in effect](#)

When items in a field are hidden by [autofiltering](#), the drop-down arrow  in the field label is blue. Click the blue arrow  to see the filter that is in effect.

[Show All](#)

Apply a filter (MDB)

[Apply a filter in a table, query, or form](#)

- Click **Apply Filter**  on the toolbar.

Notes

- When you apply a filter to a datasheet, Microsoft Access also applies filters you created in any [subdatasheet](#) within it.
- Access can apply a filter automatically when you open a form if you set the form's **OnOpen** property to a macro that uses the ApplyFilter action or to an event procedure that uses the **ApplyFilter** method of the **DoCmd** object.
- If you create a filter in a form, subform, datasheet, or subdatasheet that already contains a filter from a previous working session, the new filter replaces the previous one.

[Apply an existing query as a filter on a form or datasheet](#)

While you can save only one filter for each form, subform, datasheet, or [subdatasheet](#), you can use queries that you've already created as additional filters.

1. Click in the form, subform, datasheet, or subdatasheet, you want to filter.
2. Switch to the [Filter By Form](#) window or the [Advanced Filter/Sort window](#).
3. On the toolbar, click **Load From Query** .
4. In the **Filter** box, select a query, and then click **OK**.
5. Click **Apply Filter**  on the toolbar.

[Apply an inherited filter in a report](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is [selected](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **FilterOn** property box, click **Yes**.

Note Access can apply a filter automatically when you open a report if you set the report's **OnOpen** property to a macro that uses the `ApplyFilter` action or to an event procedure that uses the `ApplyFilter` method of the `DoCmd` object. You can also automatically apply a filter to a report using the `OpenReport` action or its corresponding method.

 [Apply an existing filter to a data access page in Page view](#)

You can reapply a filter that you created during the current working session.

1. Open a data access page in [Page view](#).
2. Do one of the following:
 - On an ungrouped page, click **Filter Toggle**  on the record navigation toolbar so that it is selected.
 - On a [grouped page](#), expand the group that you want to filter, and then click **Filter Toggle** on the record navigation toolbar for that group so that the button is selected.

Note This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

 [Reapply a filter to retrieve the most current set of records](#)

If you're working with filtered data in a [multiuser \(shared\) database](#), you may want to reapply the filter occasionally to see which records have been added, deleted, or changed since you opened the form or datasheet and applied the filter.

- With the filtered records displayed in a form, subform, datasheet, or [subdatasheet](#), click **Apply Filter/Sort** on the **Records** menu.

[Show All](#)

Create a filter (MDB)

There are four methods you can use to [filter](#) records in a form or datasheet: [Filter By Selection](#), [Filter By Form](#), [Filter For Input](#), and [Advanced Filter/Sort](#). You can also filter records in a [data access page](#).

[Filter records by selecting values in a form or datasheet \(Filter By Selection\)](#)

1. In a field on a form, subform, datasheet, or [subdatasheet](#), find one instance of the value you want records to contain in order to be included in the filter's results.
2. Select all or part of a value in a field by doing one of the following:

Note How you select the value determines what records the filter returns.

[Find records in which the entire contents of that field match the selection](#)

- Select the entire contents of a field or place the insertion point in a field without selecting anything.

For example, select the value "Berlin" in the City field to return all records with Berlin as the city.

[Find records in which the value in that field starts with the same characters you selected.](#)

- Select part of a value starting with the first character in a field.

For example, Select only "Fran" in the CompanyName field with the value "France restauration" to return all records that have a company name starting with "Fran," such as "Franchi S.p.A." and "Frankenversand".

[Find records in which all or any part of the value in that field contains](#)

[the same characters you selected.](#)

- Select part of a value starting after the first character in a field.

Select the letters "Del " in the CompanyName field with the value "Old World Delicatessen" to return all records that have "del" anywhere in the CompanyName field, such as "Ernst Handel", "Galería del gastrónomo", and "Que Delícia".

3. Click **Filter By Selection**  on the toolbar.
4. If you want to refine your search, you will need to reset the filter to show all of the records and then repeat steps 2 and 3. To reset the filter, click **Remove Filter**  on the toolbar.

Note You can also filter for records that do not have a certain value. After selecting a value, right-click it, and then click **Filter Excluding Selection**.

 [Filter records by entering values in a blank view of your form or datasheet \(Filter By Form\)](#)

1. Open a form in [Form view](#), or a table, query, or form in [Datasheet view](#).

To filter records in a [subdatasheet](#), display the subdatasheet by clicking its [expand indicator](#).

2. Click **Filter By Form**  on the toolbar to switch to the Filter By Form window.

You can specify [criteria](#) for the form, subform, main datasheet, or any subdatasheet that's displayed. Each subform or subdatasheet has its own **Look For** and **Or** tabs.

3. Click the field in which you want to specify the criteria that records must meet to be included in the filtered set of records.
4. Enter your criteria by selecting the value you're searching for from the list in the field (if the list includes field values), or by typing the value into the field.

[Find records in which a check box, toggle button, or option button is or is not selected](#)

- Click the check box or button until it's the way you want. To return it to a neutral position so that it won't be used as criteria for filtering records, continue clicking the check box or button until it's grayed.

[Find records in which a particular field is empty or not empty](#)

- Type **Is Null** or **Is Not Null** into the field. (You can select these options from the list in fields with a Memo, OLE Object, or Hyperlink data type, and in calculated fields in queries.)

[Find records using a criteria expression](#)

- Type the [expression](#) into the appropriate field or enter one using the Expression Builder.

If you specify values in more than one field, the filter returns records only if they contain the same values you specified in each of those fields.

5. To specify alternative values that records can have to be included in the filter's results, click the **Or** tab for the form, subform, datasheet, or subdatasheet you're filtering, and enter more criteria.

The filter returns records if they have all the values specified on the **Look For** tab, or all the values specified on the first **Or** tab, or all the values specified on the second **Or** tab, and so on.

6. Click **Apply Filter**  on the toolbar.

[Filter records by entering criteria in place in a form or datasheet \(Filter For Input\)](#)

1. Open a form in [Form view](#), or a table, query, or form in [Datasheet view](#).

To filter records in a [subdatasheet](#), first click the [expand indicator](#) to display that subdatasheet.

2. Do one of the following:

[Specify the exact field value you want the filtered records to contain](#)

- Right-click in the field in the form, subform, datasheet, or subdatasheet you're filtering, and then in the **Filter For** box on the shortcut menu, type the value.

For example, to find all the records that have "London" in the City field, right-click in the City field, and then type **London** in the **Filter For** box.

[Specify more complex criteria](#)

- Type the complete [expression](#) using the appropriate combination of identifiers, operators, wildcard characters, and values to produce the result you want.

For example:

- To find all the records where the Contact Title field includes the word "Marketing" anywhere in the title, right-click in the Contact Title field, and then type ***Marketing*** in the **Filter For** box.
- To display only the records for orders that were shipped more than 15 days after the specified required date, type **[ShippedDate]-[RequiredDate]>15**. (In this example, you don't have to right-click a specific field to enter the expression. However, in a datasheet with subdatasheets, first click in the sheet you want to filter.)

3. Do one of the following:

- To apply the filter and close the shortcut menu, press ENTER.
- To apply the filter and keep the shortcut menu displayed (so that you can specify additional criteria for the field), press TAB. Enter new criteria, and then press TAB again. Repeat until you have just the records you want.

[Filter records by using the Advanced Filter/Sort window](#)

1. Open a form in [Form view](#), or a table, query, or form in [Datasheet view](#).
2. Click in the form, subform, datasheet, or [subdatasheet](#), you want to filter.

3. On the **Records** menu, point to **Filter**, and then click **Advanced Filter/Sort**.
4. Add to the design grid the fields you'll need to specify the values or other criteria the filter will use to find records.
5. To specify a sort order, click in the **Sort** cell for a field, click the arrow, and select a sort order.

Microsoft Access first sorts the leftmost field in the design grid, then it sorts the next field to the right, and so on.

6. In the **Criteria** cell for the fields you have included, enter the value you're looking for or enter an [expression](#).
7. Apply the filter by clicking **Apply Filter**  on the toolbar.

 [Create a filter in a data access page](#)

 [Create a filter to limit records in a Page before they are retrieved from the database](#)

 [Create a filter on an ungrouped data access page](#)

1. Open the data access page in [Design view](#).
2. If the fields you want to display aren't in the [section](#) already, add them to the page.
3. Add the field you want to use for filtering as a text box, bound span, drop-down list box, or list box.
4. Right-click the filter field control and click **Group Filter Control**.
5. If necessary, customize the [record navigation control](#). For example, on a page that finds one record at a time, not all of the buttons are required, so you might want to show only the **New**, **Delete**, **Save**, **Undo**, and **Help** buttons on the record navigation toolbar.

Notes

- When you create a group filter control, the **GroupFilterField** and **GroupFilterControl** properties are automatically set for the [group level](#). If you convert a text box control or a [bound span control](#) to a group filter control, it will change to a drop-down list box control and the following control properties will be set: **ListRowSource**, **ListBoundField**, and

ListDisplayField.

- If you add a group filter control to a group level that already has one, the previous control will change to a simple drop-down list box control.
- To convert a group filter control to a regular control, right-click the control and click **Group Filter Control**. The control will change to a regular drop-down list box.

[Create a filter on a grouped data access page](#)

1. Open the data access page in [Design view](#).
2. If the fields you want to display aren't in the [section](#) already, add them to the page.
3. Add the field you want to use for filtering as a text box, bound span, drop-down list box, or list box.
4. Select the filter field control and do one of the following:
 - To group only on that field, click **Promote**  on the toolbar.
 - To group on the whole table or [query](#) that contains that field, click **Group by Table**  on the toolbar.

Note If you used the Page Wizard to create the page, the **Group by Table** option will not be available.

Microsoft Access adds a [group header](#) with an [expand control](#) and a [record navigation section](#) with a [record navigation control](#). If you grouped on one field, Access moves the control on which you grouped to the group header. If you grouped on a table or query, Access moves all the controls — including controls bound to fields that are in that table or query and controls bound to [Lookup fields](#)— to the group header.

5. Right-click the filter field control, and then click **Group Filter Control**.

The following steps are optional. They remove features that were added by default for a grouped page but aren't necessary when using the group filter control. Do one or more of the following:

- Select the expand control in the header that contains the filter control, and then press DELETE.

- Select the record navigation section for the [group level](#) that contains the filter control, and then press DELETE. Access deletes the section and the record navigation control that it contains.
- Customize the record navigation control for the lower group level. For example, hide the buttons that add and delete records.

Notes

- If the group header also contains a control that calculates a total or other aggregate, when you choose a group from the group filter, the calculated control displays the correct total or other aggregate for that group of records.
- You can have multiple group levels, each with one group filter. For example, you can create a group filter for a Country field, and then in the next lower group level, create a group filter for the City field, and so on.
- When you create a group filter control, the **GroupFilterField** and **GroupFilterControl** properties are automatically set for the group level. If you convert a text box control or a [bound span control](#) to a group filter control, it will change to a drop-down list box control and the following control properties will be set: **ListRowSource**, **ListDisplayField**, and **ListBoundField**.
- If you add a group filter control to a group level that already has one, the previous control will change to a simple drop-down list box control.
- To convert a group filter control to a regular control, right-click the control and click **Group Filter Control**. The control will change to a regular drop-down list box.

[Create a filter in a Page to limit records after they are retrieved from the database](#)

[Create a filter in Page view to limit records after they've been retrieved](#)

1. Open a data access page in [Page view](#).
2. Do one of the following:

[In an ungrouped page](#)

1. Move focus to the field that has the value that you want to use to filter records.
2. Click **Filter By Selection**  on the record navigation toolbar.

[In a grouped page](#)

1. Expand the group that you want to filter.
2. Move focus to the field that has the value that you want to use to filter records.
3. Click **Filter By Selection**  on the record navigation toolbar for that group.

The filter applies only to the current group. In other words, if a page groups customer orders by country/region, and you filter the records for Brazil that have a required date of 6/2/98, that filter applies only to the records for Brazil, not to the records for the other countries/regions.

3. To further narrow the subset of records, continue selecting fields and clicking **Filter By Selection**  until you have just the records you want.

For example, if you want to see records only for customers in Germany with a required date of 5/12/98, select "Germany" in the Country field, click **Filter By Selection**, select "5/12/98" in the Required Date field, and then click **Filter By Selection** again.

Notes

- If you select only part of the value in a field, the data access page still uses the entire field value for filtering.
- You can remove the filter and then reapply it during the same working session.
- This procedure doesn't apply to a [PivotTable list](#), [Spreadsheet Component](#), or [Chart Component](#). These features can be filtered independently from the rest of the data on the data access page and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Create a hyperlink that filters the records on a data access page that the link jumps to](#)

1. Open a data access page in [Design view](#).
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **Page in this database** under **Link to**.
4. In the **Text to display** box, type the text that you want to be displayed in the field or text box. If you leave the box blank, Microsoft Access will use the hyperlink address as the display text.
5. Click **ScreenTip** and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address as the ScreenTip.
6. From the list, select the data access page you want to link to.
7. To create a filter for the data access page, click **Server Filter** and type an [expression](#) in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** box. For example, if you have a data access page that displays order information, including the ID of the salesperson, and you want to link to a data access page with the employee record about that salesperson, type **EmployeeID=[EmployeeID]**.
8. Click **OK** twice.

Access adds the hyperlink to the data access page. To test the link, switch to [Page view](#) and click the hyperlink. Access opens the target page in your Web browser.

[Filter data in PivotTable or PivotChart view](#)

[Filter data in PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Filter a field in the row, column, filter, or detail area \(Autofiltering\)](#)

1. Click the arrow  for the field that contains the [items](#) of data that you want to filter.

2. Select the check boxes for the items you want to display, and clear the check boxes for the items you want to hide.

Notes

- To display or hide all of the data for a field, select or clear the **(All)** check box. A shaded check box indicates that some of the lower-level items are hidden. To see which lower-level items are hidden, click the expand indicator  beside the item with the shaded check box.
- When placed in the [row](#), [column](#), or [detail area](#), a field always supports multiple selection. A field in the [filter area](#) can support either single or multiple selection. To switch from multiple to single selection, on the **Filter and Group** tab of the **Properties** dialog box, clear the **Allow selecting multiple items when in filter area** check box.

[Display only data that matches a cell \(Filter By Selection\)](#)

- Right-click the cell that contains the value you want to display, and then click **Filter By Selection** on the shortcut menu.

[Filter a row or column field for top or bottom *n* values or a percentage \(Conditional filtering\)](#)

1. Select the field that has the values you want to filter.
2. Click **Properties**  on the **PivotTable** toolbar, and then click the **Filter and Group** tab.
3. Set **Display the** to **Top** or **Bottom**.
4. Set **Items** to a number in the list, or type the value you want.
5. To show *n* number of items, clear the % check box. To show *n*% of items, select the % check box.
6. Set **Based on** to a value in the list.

Notes

- When items in a field are hidden by an [autofilter](#) or a [filter by selection](#), the arrow  in the field label is blue. When items are hidden by a conditional filter, the filtered field displays a funnel icon  to the left of the arrow.
- When you apply a filter, the displayed totals are based on visible items only. You can change the setting to include hidden data.

 [Filter data in PivotChart view](#)

 [Filter a field in the category, series, or filter area \(Autofiltering\)](#)

If the [filter area](#) does not have a field, you can either add a new field or move a field from another area to the filter area.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the arrow  for the field that contains the items of data that you want to filter.
3. Select the check boxes for the items you want to display, and clear the check boxes for the items you want to hide.

Notes

- To display or hide all of the data for a field, select or clear the **(All)** check box. A shaded check box indicates that some of the lower-level items are hidden. To see which lower-level items are hidden, click the expand indicator  beside the item with the shaded check box.
- When placed in the category, series, or filter area, a field always supports multiple selection. A field in the filter area can support either single or multiple selection.

 [Filter a row or column field for top or bottom *n* values or a percentage \(Conditional filtering\)](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Select the field, [series](#), or [point](#) that has the values you want to filter.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Filter and Group** tab.
4. In the **Display the** box, click **Top** or **Bottom**.
5. Set **Items** to a number in the list, or type the value you want.

6. To show n number of items, clear the % check box. To show $n\%$ of items, select the % check box.
7. Set **Based on** to a value in the list.

Note When items in a field are hidden by an [autofilter](#), the arrow  in the field label is blue .

[Show All](#)

Modify a filter (MDB)

If a filter isn't giving you the results you need, you can modify it. You can modify a filter by using the same technique you used to create it or the technique that works best for the modifications you want to make.

[Add more criteria or change existing criteria by using Filter By Selection or Filter Excluding Selection](#)

1. To further limit a set of records that already has a filter applied, start with step 2. To change the criteria previously defined for a filter, first remove the filter if it is currently applied.

[How?](#)

Do one of the following:

- To remove a filter and show the records that were displayed previously in the table, query, or form, click **Remove Filter**  on the toolbar in [Form view](#) or [Datasheet view](#).
- To remove a filter in a [multiuser \(shared\) database](#) and see which records have been added, deleted, or changed since you opened the table, query, or form, click **Remove Filter/Sort** on the **Records** menu in Form view or Datasheet view.

Note Removing a filter doesn't delete it. You can reapply the filter by clicking **Apply Filter**  on the toolbar. When you remove a filter from a datasheet, Microsoft Access also removes any filters you applied to any [subdatasheet](#) within it.

2. In Form or Datasheet view, select a value in the form, subform, datasheet, or [subdatasheet](#) that you want to use as criteria, and then click either **Filter By Selection**  on the toolbar or right-click the value and click **Filter Excluding Selection** on the shortcut menu. Repeat this process until you have the records you want.

Note You can also switch to the Filter By Form window or Advanced Filter/Sort window to change the criteria, including specifying alternative criteria (one criteria or another).

[Add more criteria or change the existing criteria by using Filter For Input](#)

1. To further limit a set of records that already has a filter applied, start with step 2. To change the criteria previously defined for a filter, first remove the filter if it is currently applied.

[How?](#)

Do one of the following:

- To remove a filter and show the records that were displayed previously in the table, query, or form, click **Remove Filter**  on the toolbar in [Form view](#) or [Datasheet view](#).
- To remove a filter in a [multiuser \(shared\) database](#) and see which records have been added, deleted, or changed since you opened the table, query, or form, click **Remove Filter/Sort** on the **Records** menu in Form view or Datasheet view.

Note Removing a filter doesn't delete it. You can reapply the filter by clicking **Apply Filter**  on the toolbar. When you remove a filter from a datasheet, Microsoft Access also removes any filters you applied to any [subdatasheet](#) within it.

2. In Form or Datasheet view, do one of the following:
 - Right-click the field you want to specify criteria for in the main form, datasheet, subform, or [subdatasheet](#), and then in the **Filter For** box on the shortcut menu, type the value you want to use as criteria.
 - Right-click any field in the main form, datasheet, subform, subdatasheet, and specify the expression you want to use as criteria in the **Filter For** box.
3. Press TAB and repeat step 2 until you have the records you want.

Note You can also switch to the Filter By Form window or Advanced Filter/Sort window to change the criteria.

[Modify a filter by using Filter By Form](#)

You can modify the filter by adding to blank fields other values you want the filtered records to contain, deleting or changing values in a field, or making any of the following modifications.

[Clear all the fields in the Filter By Form window](#)

- Click **Clear Grid**  on the toolbar.

[Delete an Or tab at the bottom of the Filter By Form window](#)

1. Click the **Or** tab you want to delete.
2. On the **Edit** menu, click **Delete Tab**.

[Modify an inherited filter in a report](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is selected in the [Database window](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **Filter** property box, modify the SQL [WHERE](#) clause.

[Tip](#)

If you don't know [SQL](#), first change the filter in the underlying table or query. Next, switch to Design view for the table or query and display its property sheet. Finally, copy the revised WHERE clause from the table's or query's **Filter** property to the **Filter** property for the report.

[Show All](#)

Remove a filter (MDB)

Do one of the following:

- To remove a filter and show the records that were displayed previously in the table, query, or form, click **Remove Filter**  on the toolbar in [Form view](#) or [Datasheet view](#).
- To remove a filter in a [multiuser \(shared\) database](#) and see which records have been added, deleted, or changed since you opened the table, query, or form, click **Remove Filter/Sort** on the **Records** menu in Form view or Datasheet view.

Note Removing a filter doesn't delete it. You can reapply the filter by clicking **Apply Filter**  on the toolbar. When you remove a filter from a datasheet, Microsoft Access also removes any filters you applied to any [subdatasheet](#) within it.

[Show All](#)

Delete or remove a filter (MDB)

Deleting a filter permanently eliminates the filter. Removing a filter does not delete the filter. You remove a filter to show all records before you applied the filter. After removing a filter, you can apply it again.

 [Delete a filter](#)

 [Delete a filter in a form or datasheet](#)

Use this procedure to delete any type of filter, regardless of what method you used to create it.

1. Click in the form, subform, datasheet, or [subdatasheet](#) whose filter you want to delete.
2. Switch to the [Advanced Filter/Sort window](#).
3. On the **Edit** menu, click **Clear Grid**.
4. Click **Apply Filter**  on the toolbar.

 [Delete an inherited filter in a report](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is [selected](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **Filter** property box, delete the SQL [WHERE clause](#).
4. Set the **FilterOn** property to **No**.

 [Remove a filter](#)

 [Remove a filter to show all records in a form or datasheet](#)

Do one of the following:

- To remove a filter and show the records that were displayed previously in the table, query, or form, click **Remove Filter**  on the toolbar in [Form view](#) or [Datasheet view](#).

- To remove a filter in a [multiuser \(shared\) database](#) and see which records have been added, deleted, or changed since you opened the table, query, or form, click **Remove Filter/Sort** on the **Records** menu in Form view or Datasheet view.

Note Removing a filter doesn't delete it. You can reapply the filter by clicking **Apply Filter**  on the toolbar. When you remove a filter from a datasheet, Microsoft Access also removes any filters you applied to any [subdatasheet](#) within it.

[Remove an inherited filter in a report](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is [selected](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **FilterOn** property box, click **No** to remove the filter.

[Remove a filter to show all records in a data access page in Page view](#)

1. Open a data access page in [Page view](#).
2. Click **Filter Toggle**  on the record navigation toolbar so that it is not selected.

Notes

- Removing a [filter](#) doesn't delete it. You can reapply a filter during the current working session by clicking the **Filter Toggle** button on the record navigation toolbar so that it is selected.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Remove a filter data in a PivotTable or PivotChart view](#)

[Remove a filter from a PivotTable](#)

When items in a field are hidden by filtering a field or filtering by selection, the arrow  in the field label is blue. When items are hidden by filtering for top or bottom values, the filtered field displays a funnel icon  to the left of the arrow.

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Remove all filters](#)

- Click the **AutoFilter** button  on the **PivotTable** toolbar so that it is not selected.

Note If you click **AutoFilter**  again to turn filtering back on, the previously set filters will be reapplied.

[Remove an autofilter, or a filter by selection, from a field](#)

- Click the blue arrow  of the field from which you want to remove the filter.
- Select the **(All)** check box and click **OK**.

[Remove a conditional filter from a field](#)

- Select the field from which you want to remove [conditional filtering](#).
- On the **PivotTable** toolbar, click **Show Top/Bottom Items** , and then click **Show All**.

Note If you don't want to retain your filters, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Remove a filter from a PivotChart view](#)

When items in a field are hidden by filtering a field or filtering by selection, the arrow  in the field label is blue.

1. Open a datasheet or form in [PivotChart view](#).
2. Do one of the following:

[Remove all filters](#)

- Click the **AutoFilter** button  on the **PivotChart** toolbar so that it is not selected.

Note If you click **AutoFilter**  again to turn filtering back on, the previously set filters will be reapplied.

[Remove an autofilter from a field](#)

1. Click the blue arrow  of the field from which you want to remove the filter.
2. Select the **(All)** check box and click **OK**.

[Remove a conditional filter from a field](#)

1. Select the field from which you want to remove [conditional filtering](#).
2. On the **PivotChart** toolbar, click **Show Top/Bottom Items** , and then click **Show All**.

Note If you don't want to retain your filters, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Show All](#)

Switch between filter windows (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can switch between windows to create or modify a [filter](#) by using the method that works best for your filtering needs, or to see a different view of an existing filter. [Filter By Form](#) and [Advanced Filter/Sort](#) have their own windows. However, you use [Filter By Selection](#) or [Filter For Input](#) directly in [Datasheet view](#) or [Form view](#). Do one of the following:

[Switch to Filter By Form from Form view or Datasheet view, or the Advanced Filter/Sort window](#)

- On the toolbar, click **Filter By Form** .

[Switch to Filter By Form from Advanced Filter/Sort](#)

- On the **Filter** menu, click **Filter By Form**.

[Switch to Advanced Filter/Sort from Form view or Datasheet view](#)

- On the **Records** menu, point to **Filter**, and then click **Advanced Filter/Sort**.

[Switch to Advanced Filter/Sort from Filter By Form](#)

- On the **Filter** menu, click **Advanced Filter/Sort**.

[Show All](#)

Create a database object from filtered records (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

 [Create a form or report from filtered records](#)

1. With the filtered records displayed, click the arrow next to the **New Object** button  on the toolbar.
2. Type a name for the query, and then click **OK**.

The new query appears in the Database window.

- To use the AutoForm or AutoReport Wizard, click **AutoForm** or **AutoReport**.
- To specify the type of form or report you want to create, click **Form** or **Report**, and then make your selection.

The new object uses as its [record source](#) the table or query in which the filter was created, and it inherits the filter.

3. Complete the design or make any other changes you want to the form or report, and then save it.

Notes

- If a [sort order](#) was specified in the underlying table or query, the new form or report will use it. Whether the filter is automatically applied depends on how you create the form or report.
 - If you create the form or report directly from filtered data while the data is displayed in an open table or query, the filter is applied automatically each time you view the report, but it is applied only the first time you view the form after creating it. (The next time you open the form, you'll apply the filter yourself.)

- If you create the form or report based on a closed table or query, the filter won't be applied in the new form or report. You apply it when you need it.
- If you subsequently change the filter in a table or query, it has no effect on a form or report that was based on a filter you created previously in that table or query.

[Create a query from a filter in a table, query, or form](#)

Instead of designing a query from scratch in query [Design view](#), you can create it automatically by saving an existing filter in a table, query, or form as a query. A query offers more flexibility than a filter.

1. With the filter displayed in the [Filter By Form window](#) or the [Advanced Filter/Sort window](#), click **Save As Query**  on the toolbar.
2. Type a name for the query, and then click **OK**.

The new query appears in the Database window. It automatically includes all the fields from the underlying table because the query's **OutputAllFields** property is automatically set to **Yes**, but you can change the query's design if necessary.

Note You can use the query as a filter, too. You might do this if you already saved one filter with the form or datasheet, and you want to use the query as a second filter.

[Show All](#)

Enable or disable filtering records in a form (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

- Set the form's **AllowFilters** property to **Yes** to enable filtering or to **No** to disable filtering capabilities in [Form view](#) and in form [Datasheet view](#).

 [How?](#)

1. Open a table, query, form, report, or [data access page](#) in [Design view](#).
2. Open the property sheet by doing one of the following:
 - On a form or report, double-click the [form selector](#) or the [report selector](#).



- 1 Form or report selector

- On a data access page, click **Select Page** on the **Edit** menu.
- On a table or query, click **Properties**  on the toolbar.
- For a section on a form or report, double-click the [section selector](#).



- 1 Section selector

- For a section on a data access page, double-click the section bar.



1 Section bar

- To set properties of a query field, click the cell in the **Field** row, and then click **Properties**  on the toolbar.
 - To set properties of a query field list, click anywhere in the list, and then click **Properties**  on the toolbar.
 - To set properties for the body of a data access page, click above the first section, and then click **Properties**  on the toolbar.
 - To set properties for a [control](#), click the control, and then click **Properties**  on the toolbar.
3. In the property sheet, click the property you want to set, and then do one of the following:
- If an arrow appears in the property box, click it and then select a value from the list.
 - Type a setting or [expression](#) in the property box.
 - If a **Build** button  appears next to the property box, click it to display a [builder](#) or list of builders.

Tips

- To get Help on any property in the property sheet, click the property and then press F1.
- If you need more space to enter or edit a property setting, press SHIFT+F2 to open the **Zoom** box.

Note If the **AllowDesignChanges** property of a form is set to **All Views**, you can also set form properties in [Form view](#) and [Datasheet view](#).

Note When you set the **AllowFilters** property to **No**, you disable [Filter By Selection](#), [Filter By Form](#), [Filter For Input](#), and [Advanced Filter/Sort](#).

[Show All](#)

Troubleshoot filters (MDB)

[The filter returns too many records.](#)

[Filter By Form](#)

- Enter a value or [expression](#), or pick a value from the list in other fields on the **Look For** tab.
- Delete some of the alternative criteria that records can meet by deleting one or more **Or** tabs.

[Filter By Selection](#)

- Select another value that you want the records to contain, and then click **Filter By Selection**  on the toolbar. Repeat this until you have just the records you want to see.

[Filter Excluding Selection](#)

- Make sure you haven't selected values in more than one row at a time. Although you can do this, Microsoft Access interprets the filter criteria as follows: "exclude the first selected value OR the second selected value," instead of "exclude the first selected value AND the second selected value."

Instead, select the first value you want to exclude, right-click, and then select **Filter Excluding Selection**. Select the next value you want to exclude, and then repeat the previous step.

- Change the filter by pointing to **Filter** on the **Records** menu, and then clicking **Advanced Filter/Sort**. In the **Criteria** row of the design grid, change **Or** to **And**.

[Filter For Input](#)

- Specify another value in the **Filter For** box, and then press ENTER. Repeat until you have the records you want.

[Advanced Filter/Sort window](#)

- Add more fields to the design grid, and then specify criteria for them in the **Criteria** row.
- Delete one or more extra rows of criteria (the **Or** rows) and use expressions with **And** instead of **Or**.

[The filter returns too few records.](#)

[Filter By Form](#)

- Add alternative criteria that records can meet on the **Or** tab (at the bottom of the window).
- If you have multiple criteria on the **Look For** or **Or** tab, delete the value or expression that's restricting the records you want to see.

[Filter By Selection](#)

- Remove the filter, select a value you're looking for, and then click **Filter By Selection**  on the toolbar. If necessary, repeat this until you have the records you want to see.
- If you're using [Filter By Selection](#) on a combo box field that gets its values from a value list, check if the bound column for the combo box is the first column displayed in the list. If it's not and you've selected as your criteria part of a value from the combo box's displayed column, Microsoft Access can't match the criteria to the data in the bound column. You need to either change the combo box so the bound column is the first column displayed in the list, or fill the combo box with values from a table or query.

[Filter For Input](#)

- Remove the filter if it's currently applied, and then specify a different value or expression in the **Filter For** box.
- Switch to the [Advanced Filter/Sort window](#) to see if you specified more than one criteria that records must meet, or if you specified the wrong criteria.

[Advanced Filter/Sort window](#)

- Delete criteria for one or more fields in the design grid.
- Add extra **Or** rows where you can specify alternative criteria.
- Use expressions with **Or** instead of **And**.

[The filter returns the wrong records.](#)

If the filter is returning the wrong records, make sure you're using the correct fields to specify the value you're searching for or to specify other criteria, and make sure you've entered the correct value or criteria. If you're filtering on fields from linked tables, the values you use in criteria for those fields are [case-sensitive](#)— they must match the case of the values in the underlying table.

- If you're using expressions in the [Filter By Form](#) window or [Advanced Filter/Sort window](#), verify you're using the correct [operators](#).
- If you're specifying multiple criteria in the Advanced Filter/Sort window, verify you've entered it in the correct rows and columns in the design grid.

[The filter didn't sort the records in the correct order.](#)

You may need to:

- Add or delete a [sort order](#).
- Sort by a different field.
- Change the sort order from **Ascending** to **Descending**, or vice versa.
- In the [Advanced Filter/Sort window](#), rearrange the columns in the design grid, and then specify a sort order on one or more fields. Microsoft Access sorts the records starting with the leftmost column in the grid.

Notes

- If you have numbers in fields with a [Text data type](#), those numbers will not sort from lowest number to highest, or vice versa. Instead, they will sort as follows: 1, 11, 12, 2, 20, 3, and so on. You must pad the single-digit numbers with a leading zero for the strings to be sorted properly.
- If you have **Null** values in a field that you're sorting in ascending order, records with those values appear first in the sort order before any records with text or numeric values in the sorted field.

[I can't use an existing query as a filter.](#)

A query must meet certain requirements if you're going to use it as a filter. The query:

- Must be based on the same underlying table or query as the form or datasheet to which you're applying it.
- Can't include other tables or queries.
- Must be a [select query](#) (not a [crosstab query](#) or [action query](#)).
- Can't include fields that sum, count, or calculate another type of total for the values in a field.

[Some of the fields in the Filter By Form window are unavailable.](#)

You can't use the following types of fields in the [Filter By Form](#) window because they don't apply to filtering records:

- A [calculated control](#) in a form.
- A field created with an [unbound control](#) or [unbound object frame](#).

[I don't have lists of values to pick from on one or more of the fields in the Filter By Form window.](#)

How many records Microsoft Access reads depends on whether or not the field is [indexed](#). If the field is indexed, Microsoft Access reads only the unique values, not all the values in a field. If the field isn't indexed, Microsoft Access reads all the values in the field. If the number of records it reads is more than the maximum it's allowed to display, which is determined by the setting for **Don't Display Lists Where More Than This Number Of Records Read**, Microsoft Access doesn't display the values for that field on the list. Consequently, you need to change the setting for this option. Because Microsoft Access reads all values in nonindexed fields, not just unique values, set this option to a number greater than (or equal to) the number of records in the nonindexed field with the most values.

Note When Microsoft Access is able to display the list in a field, it only shows the unique values, even for a nonindexed field.

[I created a filter on a subform in which the records were synchronized with the main form, but the filter didn't have any affect on the main form.](#)

The filter behaved correctly. When you create a filter on a subform, it only applies to the subform. To filter the records on the main form, click the main form, and then create the filter.

[I created a filter on a subdatasheet, but the filter didn't have any effect on the datasheet that displays the subdatasheet.](#)

The filter behaved correctly. When you create a filter on a [subdatasheet](#), it only applies to the subdatasheet. To filter the records on the datasheet that displays the subdatasheet, click in that datasheet, and then create the filter.

[I can't display my records again after using the Data Entry command.](#)

- On the **Records** menu, click **Remove Filter/Sort**.

[My wildcard characters aren't working as expected.](#)

In a Microsoft Access database, the wildcard characters can be different depending on the current setting of the ANSI SQL query mode.

[About using wildcard characters to search for partial or matching values](#)

You use wildcard characters as placeholders for other characters when you are specifying a value you want to find and you:

- Know only part of the value.
- Want to find values that start with a specific letter or match a certain pattern.

Wildcard characters are meant to be used with fields that have the Text data type. You can sometimes use them successfully with other data types, such as dates, if you don't use the Microsoft Windows Control Panel to change the regional settings for these data types.

If you are using Microsoft Jet database engine SQL in a [Microsoft Access database](#), you can use the following wildcard characters in [queries](#) and [expressions](#) to find such things as field values, records, or file names. You can also use these characters in the **Find** and **Replace** dialog boxes in an Access database or a [Microsoft Access project](#).

Character	Description	Example
*	Matches any number of characters. It can be used as the first or last character in the character string.	wh* finds what, white, and why
?	Matches any single alphabetic character.	B?ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill b[!ae]ll finds bill and bull but not ball or bell
!	Matches any character not in the brackets.	
-	Matches any one of a range of characters. You must specify the range in ascending order (A to Z, not Z to A).	b[a-c]d finds bad, bbd, and bcd
#	Matches any single numeric character.	1#3 finds 103, 113, 123

To find such things as field values, records, or file names in a Microsoft Access project, or in a Microsoft Access database that uses Microsoft SQL Server-compatible syntax, use the following ANSI-92 wildcards.

Note If you are using the **Find** and **Replace** dialog boxes in an Access project or in an Access database that uses Microsoft SQL Server-compatible syntax, you must use the Microsoft Jet SQL wildcards.

Character	Description	Example
%	Matches any number of characters. It can be used as the first or last character in the character string.	wh% finds what, white, and why
_	Matches any single alphabetic character.	B_ll finds ball, bell, and bill
[]	Matches any single character within the brackets.	B[ae]ll finds ball and bell but not bill b[^ae]ll finds bill and bull but not ball or bell
^	Matches any character not in the brackets.	
-	Matches any one of a range of characters. You	b[a-c]d finds bad,

- must specify the range in ascending order (A to bbd, and bcd Z, not Z to A).

Notes

- When using wildcard characters to search for other wildcard characters such as an asterisk (*), question mark (?), number sign (#), opening bracket ([), or hyphen (-), you must enclose the item you're searching for in brackets. If you're searching for an exclamation point (!) or a closing bracket (]), you don't need to enclose it in brackets.

For example, to search for a question mark, type **[?]** in the **Find** dialog box. If you're searching for a hyphen and other characters simultaneously, place the hyphen before or after all the other characters inside the brackets. (However, if you have an exclamation point (!) after the opening bracket, place the hyphen after the exclamation point.)

- You can't search for the opening and closing brackets ([]) together because Microsoft Access interprets this combination as a [zero-length string](#). You must enclose the opening and closing brackets in brackets ([[]]).

[Show All](#)

About sorting records (MDB)

[Simple and complex sorts](#)

There are two types of sorts you can do; simple and complex.

- **Simple sorts** When you sort in [Form view](#), [Datasheet view](#), or [Page view](#), you do a simple sort, which means you can sort all records in ascending or descending order (but you cannot use both [sort orders](#) on more than one field).
- **Complex sorts** When you sort in query [Design view](#), the [Advanced Filter/Sort window](#), report Design view, Page Design view, PivotTable view, or PivotChart view, you can perform complex sorts. This means you can sort records in ascending order by some fields and in descending order by others.

[How a sort is saved or re-applied depends on the object](#)

- **Saving or re-applying a form, datasheet, report, PivotTable, or PivotChart sort order** Regardless of where you specify the [sort order](#), Microsoft Access saves the sort order when you save the form, datasheet, report, PivotTable, or PivotChart. In a form, datasheet, PivotTable, or PivotChart, Access also reapplies the sort automatically when you reopen that object or view, or base a new form or report on a form or datasheet.
- **Saving a data access page sort order** In a Page, Access saves the sort order if you create it in Page [Design view](#) (by using Data Outline or setting the **DefaultSort** property), but doesn't save the sort order if you create it in [Page view](#) (by using the Sort buttons).

[Considerations when sorting records](#)

- If you base a new [form](#) or [report](#) on a [table](#) or [query](#) that has a [sort order](#) saved with it, the new form or report inherits the sort order.
- Access will sort on up to 255 characters, in one or more fields, in the results

- of a query or advanced filter.
- Sort order depends on the language setting you specify in the **New Database Sort Order** box (**Options** dialog box, **General** tab) at the time you create your database. If your database contains linked tables from a database that uses a different language sort order, Access uses the sort order of the database that contains the link to the table, not the database where the table is stored.
 - If the design grid for the query or filter contains the asterisk from the field list, you can't specify a sort order in the design grid unless you also add the fields you want to sort on to the design grid.
 - To sort dates and times from earlier to later, use ascending order. Use descending order to sort from later to earlier.
 - Numbers stored in [Text](#) fields are sorted as strings of characters, not numeric values. Therefore, to sort them in numeric order, all text strings must be the same length with shorter numbers padded with leading zeros. For example, the result of an ascending sort of the text strings "1", "2", "11", and "22" will be "1", "11", "2", "22". You must pad the single-digit numbers with a leading zero for the strings to be sorted properly: "01", "02", "11", "22". For fields that don't contain **Null** values, another solution would be to sort on the numeric values of the strings by using the **Val** function. For example, if the Age column is a Text field that contains numeric values, specifying Val([Age]) in a **Field** cell and specifying a sort order in its **Sort** cell will put the records in the correct order. If you are only storing numbers or dates in a Text field, consider changing the field's data type to [Number](#), [Currency](#), or [Date/Time](#) in the table where the field is stored. Then, when you sort on the field, the numbers or dates will sort in numeric or date order without requiring leading zeros.
 - When you sort a field in ascending order, any records in which that field is blank (contains a **Null** value) are listed first. If a field contains records with both **Null** values and [zero-length strings](#), the fields with **Null** values appear first in the sort order, immediately followed by the zero-length strings.

[Sorting on fields with a data type of Memo, Hyperlink, or OLE object](#)

You can't sort an [OLE Object](#) under any circumstances. Although you can't sort a [Memo](#) or [Hyperlink](#) field in a Page, you can sort these fields in a table, query, form, or report. In a report, you must type the name of the Memo or Hyperlink field into the **Field/Expression** box of the **Sorting and Grouping** dialog box because those fields don't show up in the list of available fields.

Note Memo fields will only be sorted based on the first 255 characters.

[About sorting data in PivotTable or PivotChart view](#)

[In PivotTable view](#)

[Sort in ascending or descending order](#)

For example, you can sort a sales column in ascending order or an employee name column in descending order.

You can also sort data separately for each column. The order in which you sort the columns determines how the data will be organized.

Salesperson	Product	
Buchanan	Beverages	
Buchanan	Dairy	
Buchanan	Meat	
Buchanan	Produce	
Davolio	Beverages	
Davolio	Meat	
Davolio	Produce	
Suyama	Beverages	
Suyama	Dairy	
Suyama	Meat	
Suyama	Produce	

In this example, data is organized by salesperson. The Product column was sorted first so that the products would be in alphabetical order; the Salesperson column was sorted second.

When sorting multiple columns, first determine the order in which you want to sort the columns, and then work backward, sorting the outermost column last. For example, if you're sorting two columns, determine which is the outer column and which is the inner column. Sort the inner column first, and then sort the outer column. To list the products in alphabetical order for each salesperson, you would first sort the inner Product column and then sort the outer Salesperson column.

[Define a custom sort order](#)

You can define a custom order for a [row](#) or [column field](#) when sorting in ascending or descending order does not meet your requirements. For example,

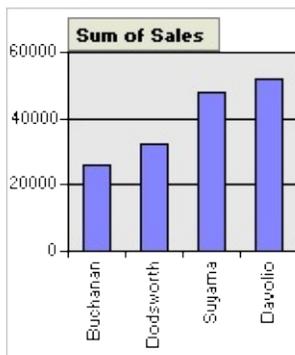
you might want to show the values in the Title column ordered on the basis of the seniority of the title. If the column had three values— Vice President, General Manager, and Manager— sorting in ascending or descending order will not show the data in the order you want. In this case, you might want to explicitly define the order of the items for the Title column.

If items are added to a field after you define a custom sort order, the new items will appear at the bottom of the field until you rearrange them.

 [In PivotChart view](#)

In a PivotChart view, you can sort numerical data or alphabetical data in ascending or descending order. For example, you can sort a [series](#) that contains numerical sales data to show the highest to lowest sales amounts, or you can sort a [category field](#) that contains company names to display in ascending order (A - Z).

In the following example, the [series field](#), which consists of sales amounts, is sorted in ascending order to show lowest to highest sales. If the category field had been sorted in ascending order, the salespeople's names would be displayed on the category axis in alphabetical order instead.



If you have a stacked bar, area, or column chart, you can sort numerically based on the height of the entire stacked value rather than just one series.

[Show All](#)

Sort records (MDB)

[Sort records in Form view or Datasheet view](#)

1. Click the field you want to use for sorting records. To sort records in a subform, click the field you want to sort. To sort records in a [subdatasheet](#), display the subdatasheet by clicking its [expand indicator](#), and then click the field you want to sort.
2. Click **Sort Ascending**  or **Sort Descending** .

Note In a form, you can sort on only one field at a time. In [Datasheet view](#), when you sort the subdatasheet for one record, Microsoft Access sorts all the subdatasheets at that level. In a datasheet or subdatasheet, you can select two or more adjacent columns at the same time, and then sort them. Access sorts records starting with the leftmost selected column. When you save the form or datasheet, Access saves the [sort order](#).

[Sort records by using the design grid](#)

1. Open a query in [Design view](#) or display the [Advanced Filter/Sort window](#) for a table, query, or form.
2. To sort on more than one field, first arrange the fields in the [design grid](#) in the order you want the sorts performed. Access sorts on the leftmost field first, then on the next field to the right, and so on. For example, to sort on the LastName field first and then on the FirstName field, the LastName field must be to the left of the FirstName field in the grid.
3. In the **Sort** cell for each of the fields you want to sort on, click an option.
4. To see the results of the query, click **View**  on the toolbar.

[Sort records in a report](#)

[Specify a sort order](#)

1. Open the report in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And**

Grouping box.

3. In the first row of the **Field/Expression** column, select a field name, or type an [expression](#).

The field or expression in the first row is the first sorting level (the largest set). The second row is the second sorting level, and so on.

Note When you fill in the **Field/Expression** column, Microsoft Access sets the **Sort Order** to **Ascending**.

Notes

- You can sort on up to 10 fields or expressions in a report.
- The Customer Labels report in the Northwind sample database sorts records by the values in three fields. To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Customer Labels report in Design view.

[Reapply an inherited sort order](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is [selected](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **OrderByOn** property box, click **Yes** to reapply it.

Note If you specify a sort order in a report wizard or in the **Sorting And Grouping** dialog box in report Design view, it overrides the inherited sort order.

[Sort records on a data access page](#)

[Sort records on a page](#)

You can sort data on a data access page, including data that's grouped. For grouped data access pages, you can sort the list of groups or the records within a group.

1. Open a data access page in [Page view](#).

2. Do one of the following:
 - In an ungrouped page, click the field you want to use for sorting records.
 - In a grouped page, expand the group that you want to sort, and then click the field you want to sort records on.

Note When you sort records within a group, the sort order applies only to that group. It does not affect the order of records in other groups.

3. Do one of the following:
 - To sort in ascending order, click **Sort Ascending** .
 - To sort in descending order, click **Sort Descending** .



If the records are displayed in tabular format, simply click the label of the field you want to sort records on. The records will display in ascending order. To sort records in descending order, click the label again.

Notes

- When you close the data access page, you'll lose the sort order. You can sort the records again when you open the page.
- The order in which numbers, text, and special characters are sorted depends on the database and your regional settings in Microsoft Windows Control Panel.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be sorted independently from the rest of the data on the data access page, and may also support additional sorting capabilities. For more information, click **Help**  on the toolbar for the PivotTable list, spreadsheet, or chart, and then search for "sorting."



1. Open the data access page in [Design view](#).
2. Scroll to a [section](#) that belongs to the group for which you want to specify the sort order.

3. Right-click the [section bar](#), and then click **Group Level Properties**.
4. In the **DefaultSort** property box, type the name of the field you want to sort on. If you want to sort on multiple names, separate each field name with a comma. If you want to sort in ascending or descending order, follow each name with a space and the keyword ASC or DESC, respectively. For example, type **OrderDate DESC, OrderID ASC**. If you don't specify the order, Microsoft Access sorts in ascending order.

Note If the control you want to sort on is bound to an [expression](#), you must type the alias of the control as displayed in its **ControlSource** property. For example, if the **ControlSource** property of the control you want to sort on is set to **GroupOfExtendedPrice: ExtendedPrice**, type **GroupOfExtendedPrice** in the **DefaultSort** property box.

[Sort data in PivotTable or PivotChart view](#)

[Sort data in ascending or descending order](#)

You can sort data that is displayed in natural or [custom order](#) in ascending or descending order.

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. Do one of the following:

[In PivotTable view](#)

- To sort the data in a field based on the [item](#) labels, click the label of the field that contains the data you want to sort.
- To sort the data in a field based on the summary values in a column, click a cell in the column that contains the values that you want to sort on.
- To sort the data in a field based on subtotal or grand total values, click a subtotal or grand total value for the field that contains the data you want to sort.

[In PivotChart view](#)

- To sort numeric data in a series, click the [series](#) or a [series point](#) within the series you want to sort.
 - To sort items in a field alphabetically, click the [category field](#) or [series field](#), or click a category axis label.
3. On the **PivotTable** or **PivotChart** toolbar, click **Sort Ascending**  or **Sort Descending** .

Notes

- To sort on multiple fields, start by sorting the innermost field first and work backward, sorting the outermost field last.



[Specify a custom sort order in PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Select the item you want to move.
3. Drag the selected item and drop it over the item above which you want to place it.
4. Repeat steps 2 and 3 until you have the items in the order you want.

Notes

- To switch from ascending or descending sort to custom sort, click **Sort Ascending**  or **Sort Descending**  on the **PivotTable** toolbar so that the button is not selected.
- If a field with custom sorting has a parent field, each parent member will display the lower-level members in custom order.

Note The order in which numbers, text, and special characters are sorted depends on the source data and your regional settings in Microsoft Windows Control Panel.

[Show All](#)

Delete a sort (MDB)

[Delete a sort order in Form view or Datasheet view](#)

- Click **Remove Filter/Sort** on the **Records** menu.

Note If you specified the sort order in the Advanced Filter/Sort window where you also specified criteria for a filter, you can delete the sort order in the **Sort** cell in the design grid and still leave the filter's criteria intact.

[Delete an inherited sort order in a report](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is selected in the [Database window](#).
2. Click **Properties**  on the **Report Design** toolbar to display the report's property sheet.
3. In the **OrderByOn** property box, click **No** to remove the sort order.

Note If you specify a sort order in a report wizard or in the **Sorting And Grouping** dialog box in report Design view, that sort order overrides the inherited sort order.

[Show All](#)

Troubleshoot sort (MDB)

[The records the query retrieves are in the wrong order.](#)

- Specify a [sort order](#) if you haven't already done so.

If you're working in a [multiuser](#) environment, someone else may have set a different sort order. Sort orders are saved with the query. You can change the sort order to the one you want.

- If you've already specified a sort order, it may be set for the wrong field. Delete the sort you specified and specify a new sort order for the correct field.
- If the sort order is set for the correct field, you may have selected **Ascending** when you really wanted **Descending**, or vice versa. Select the correct sort order.
- If you have sort orders specified for more than one field, you may need to reorder the fields in the query design grid because Microsoft Access performs multiple sorts from left to right in the order the fields appear in the query design grid.
- The field you specified the sort order for may be hidden. If you want to show the field you're using to sort records, make sure the **Show** check box is selected in the query design grid.
- You can't set a sort order on a [Hyperlink](#) (or [Memo](#)) field in a [data access page](#). However, you can sort on an [expression](#) by using the **HyperlinkPart** function.

[I can't find the language I want in the New database sort order box.](#)

Some languages are included in the [General sort order](#) setting, although they used to be listed separately in the **New Database Sort Order** box (**Tools** menu, **Options** dialog box, **General** tab). The general sort order includes the following languages: Afrikaans, Albanian, Arabic, Basque, Bulgarian, Byelorussian, Catalan, Dutch, English, Faeroese, Farsi, German - Standard, Greek, Hebrew, Hindi, Indonesian, Italian, Malay, Portuguese, Russian, Serbian, Swahili, and Urdu.

[Access isn't using the sort order I specified in a linked table.](#)

If your database contains linked tables from a database that uses a different language [sort order](#), Access uses the sort order of your database, not the database where the table is stored.

[I receive replication errors after I change the language sort order.](#)

If you're [synchronizing](#) between [replicas](#) in an Access database, you might encounter synchronization conflicts if the databases don't have the same language [sort orders](#).

[Show All](#)

About grouping records

[On a report](#)

You can group records in a report based on the values in one or more fields. For example, to look at all the orders shipped on a specific date, you could group records on the values in the ShippedDate field and sort on the values in the Country and Company fields.

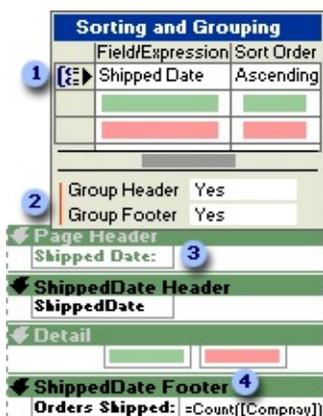


A new group is started for each date, so you can skim the report to quickly find the records for a particular date.

You can also calculate totals and other values for each group. This report prints the number of orders shipped each day.

[Setting group properties](#)

You create groups by setting properties in the **Sorting and Grouping** dialog box. You can set properties to display a header and footer for the group.



- 1 This symbol indicates that records are grouped on ShippedDate.
- 2 You create a group by setting either **GroupHeader** or **GroupFooter** or both to **Yes**.
- 3 Place a text box that identifies the group in the new [group header](#).
- 4 Place a text box that calculates a total for the group in the [group footer](#) section.

[Grouping levels](#)

You can group on up to 10 fields or expressions in a report. When you group on more than one field or expression, Microsoft Access nests the groups according to their [group level](#). The first field or expression you group on is the first and most significant group level; the second field or expression you group on is the next group level; and so on.

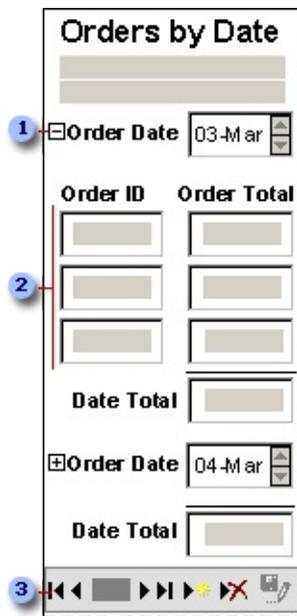


- 1 A header on a group level ...
- 2 ... is paired with a footer on the same level.

[On a data access page](#)

Grouping records on a data access page is similar to grouping records in a report: You can create a hierarchy that groups records from general categories to specific details.

The following illustration shows how the sample data access page, Orders by Date—which shows order totals grouped by date—looks in [Page view](#) or Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later .



- 1 Click the expand indicator ...
- 2 ... to display the data in the next lower group.
- 3 Use the record navigation toolbar to move between records, to sort and filter records, and to get Help.

[Parts of a grouped data access page](#)

[Body](#)

The body is the basic design surface of a data access page. On a grouped page, you can use the body to display informational text and sections. By default, the positions of text, sections, and other elements in the body are relative. That is, elements flow one after another in the same order as they appear in the HTML source. The position of an element in the body is determined by the preceding content. When viewed in [Page view](#) or Microsoft Internet Explorer, the content in the body adjusts itself to fit the size of the Web browser.

[Sections](#)

Use [sections](#) to display text, data from a database, and toolbars. The position of controls and other elements within a section is absolute by default. That is, the position of each control or element is fixed relative to the top and left

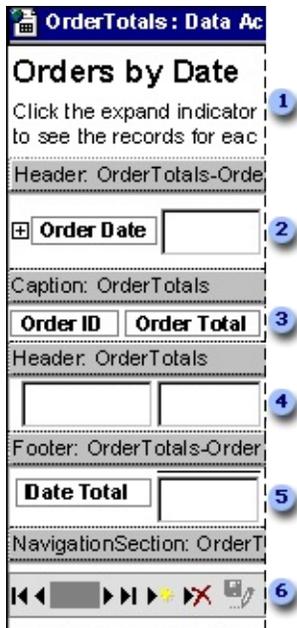
coordinates of the section. Controls that have an absolute position within a section keep the same position even when you resize the browser.

Four types of sections are available on a grouped data access page: group header, group footer, caption, and record navigation. You can use just the sections you need for a [group level](#).

- The group header is used primarily to display data and calculate totals. To group data, you must have at least two group levels. The group header at the lowest group level is like the detail section of a report: It repeats until all the records are printed for the current group.
- The group footer is used primarily to calculate totals. It appears before the record navigation section for the group level. A group footer is not available for the lowest group level on a data access page.
- The caption section is used to display captions for columns of data. It appears immediately before the group header. This section is displayed only when the next-higher group level is expanded. You can't place [bound controls](#) in a caption section.
- The record navigation section is used to display the record navigation control for the group level. A record navigation section for a group appears after the group header if there is no footer for the group level, and appears after the footer if there is a footer for the group level. You can't place bound controls in a record navigation section.

 [A grouped data access page in Design view](#)

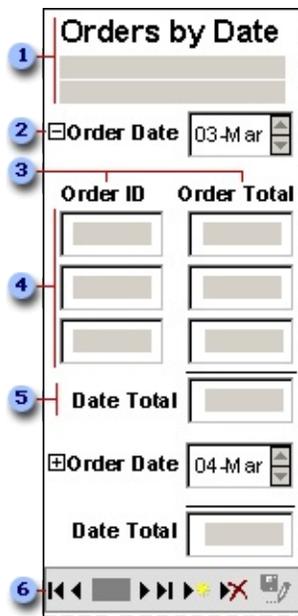
Each [group level](#) has a [record source](#). The name of the record source is displayed on the [section bar](#) for each section used for a group level. The following illustration shows the Orders by Date data access page in [Design view](#).



- 1 Page title and instructions in the body of the page
- 2 Group header for the OrderTotals-OrderDate group level
- 3 Caption section for the OrderTotals group level
- 4 Group header for the OrderTotals group level
- 5 Group footer for the OrderTotals-OrderDate group level
- 6 Record navigation section for the OrderTotals-OrderDate group level

 [A grouped data access page in Page view or Internet Explorer](#)

The following illustration shows how the sections look on the same data access page when it's open in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later. You can specify how many records you want displayed on the page for each [group level](#). In the example page, the outer group level (OrderTotals-OrderDate) shows two records at a time, whereas the inner group level (OrderTotals) shows all the records.



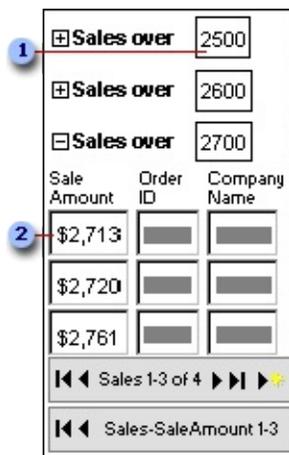
- 1 Title and instructions in the body of the page
- 2 Group header for the OrderTotals-OrderDate group level
- 3 Caption section for the OrderTotals group level
- 4 Group header for the OrderTotals group level
- 5 Group footer for the OrderTotals-OrderDate group level
- 6 Record navigation section for the OrderTotals-OrderDate group level

[About grouping records on intervals of autonumbered, currency, or numeric values on a page](#)

When you group records by autonumbered, currency, or numeric values, the control you group on must be bound to a field with one of the following data types: Autonumber, Currency, or Numeric in a [Microsoft Access database](#); decimal, numeric, int, or any other data type that supports autonumbered, currency, or numeric values in a [Microsoft Access project](#).

[How a page that groups records on intervals of \\$100 looks in Page view or in Microsoft Internet Explorer](#)

To group records on an interval, you need two controls bound to the field you want to group on. You use one control to group the records and display the field values as intervals. You use the other control, which is placed in the next lower [group level](#), to display the full value of the field. In the following example, the records come from a [query](#) named Sales. The records are grouped in intervals of 100 in a control bound to the SaleAmount field. In the next lower group level, the page displays, and sorts the records by, the actual values in the SaleAmount field.



1 This control, bound to the SaleAmount field, shows the values in the field in intervals of \$100.

2 This control, bound to the SaleAmount field, displays the actual values in the field.

[How a page that groups records on intervals of \\$100 looks in Design view](#)

The following illustration shows the layout of the example page in [Design view](#).



- 1 The control that groups records in intervals of \$100 is in the first, or highest, group level.
- 2 The control that displays the actual sale amount is in the next lower group level.

 [Property settings](#)

The following table lists the properties that are set for the [group levels](#).

Property	Sales-SaleAmount	Sales
GroupInterval	100	1.0
GroupOn	dscInterval	dscEachValue
DefaultSort	GroupOfSaleAmount ASC	SaleAmount ASC

Notes

- When you group on a control, Access creates an alias for the control by adding the prefix "GroupOf" to the control's name in the **ControlSource** property; for example, GroupOfSaleAmount: SaleAmount. When you reference this control in an expression or property, you must use the alias.
- In the **DefaultSort** property, ASC stands for ascending order and DESC stands for descending order.

 [About grouping records on intervals of date or time values on a page](#)

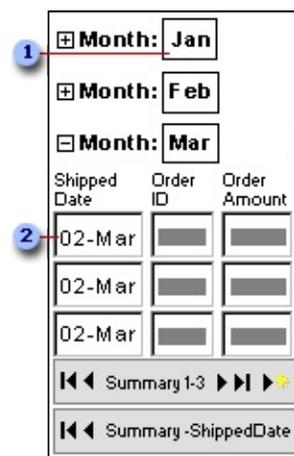
When you group records on date or time intervals, the control you group on must

be bound to a field with one of the following data types: Date/Time in a [Microsoft Access database](#); datetime, smalldatetime, or any other data type that stores date and time values, in a [Microsoft Access project](#).

[How a page that groups records by month looks in Page view or in Microsoft Internet Explorer](#)

To group records on an interval of date or time values, you need two controls that are bound to the field that contains the date or time values. One of them is used to group the records and is hidden. The other control is placed in the next lower [group level](#) and shows the individual values. In the group level that contains the hidden control, you also create a control that is bound to an [expression](#) that returns the month equivalent of the dates grouped in the hidden control.

The following illustration shows the page in [Page view](#). The hidden control is not shown in the illustration.



- 1 This control is bound to an expression that returns the month portion of the values in the ShippedDate field.
- 2 This control, bound to the ShippedDate field, sorts the records by shipped date.

[How a page that groups records by month looks in Design view](#)

In [Design view](#) of the example page, you can see the hidden control that is used to group the records in the first [group level](#).



1 This control, visible in Page view or Microsoft Internet Explorer, is bound to an expression that returns the month portion of the values in the ShippedDate field.

2 This control, not visible in Page view or Internet Explorer, is bound to the ShippedDate field and is used to group the records by month.

3 This control, also bound to the ShippedDate field, sorts the records by shipped date.

[Property settings](#)

The following table lists the properties that are set for the [group levels](#).

Property	Sales-SaleAmount	Sales
GroupInterval	1.0	1.0
GroupOn	dscMonth	dscEachValue
DefaultSort	GroupOfShippedDate ASC	ShippedDate ASC

Notes

- When you group on a control, Access creates an alias for the control by adding the prefix "GroupOf" to the control's name in the **ControlSource** property; for example, GroupOfShippedDate: ShippedDate. When you reference this control in an expression or property, you must use the alias.
- In the **DefaultSort** property, ASC stands for ascending order and DESC stands for descending order.

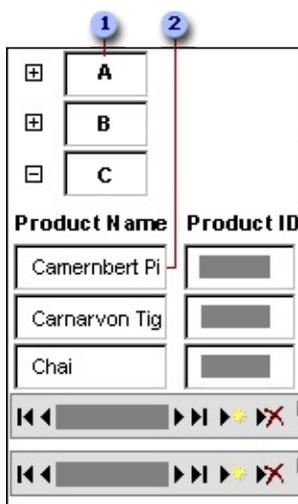
[About grouping records on the first \$n\$ characters in a text value on a page](#)

When you group records on the first n (or prefix) characters in a text field, the control you group on must be bound to a field of one of the following data types: Text in a [Microsoft Access database](#); char, varchar, nchar, or any other data type that can store text, in a [Microsoft Access project](#).

[How a page that groups records on the first character in a field looks in Page view or in Microsoft Internet Explorer](#)

To group records on the first n characters, you need two controls bound to the field you want to group on. You use one control to group the records by the specified number of prefix characters, and then display the prefix characters. You use the other control, which is in the next lower [group level](#), to sort records and display the actual value of the field.

In the following example, the records come from the Products table. In the first group level, the records are grouped on the first character of the values in the ProductName field. In the next lower group level, the page displays and sorts the records by the actual values in the ProductName field.



- 1 This control is bound to the ProductName field and shows just the first character of the field's values.
- 2 This control, also bound to the ProductName field, displays the actual values in the field.

[How a page that groups records on the first characters in a field looks in Design view](#)

The following illustration shows the layout of the example page in [Design view](#).



- 1 The control that groups records by the first character of each product name is in the first, or highest, [group level](#).
- 2 The control that displays the actual product names is in the next lower group level.

[Property settings](#)

The following table lists the properties that are set for the [group levels](#).

Property	Products-ProductName	Products
GroupInterval	1.0	1.0
GroupOn	dscPrefix	dscEachValue
DefaultSort	GroupOfProductName ASC	ProductName ASC

Notes

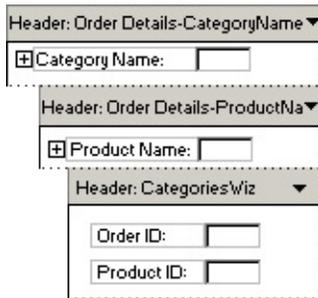
- When you group on a field, Access creates an alias for the control by adding the prefix "GroupOf" to the control's name in the **ControlSource** property; for example, GroupOfProductName: Product Name. When you reference this control in an expression or property, you must use the alias.
- In the **DefaultSort** property, ASC stands for ascending order and DESC

stands for descending order.

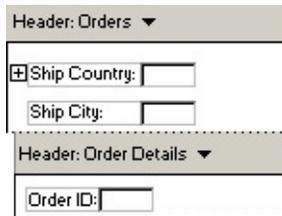
[About grouping records on multiple fields](#)

You can group records on a page based on more than one field. You can either create nested [group levels](#) or add the grouping fields to a single group level.

The following illustration shows a data access page that has three group levels in [Design view](#).



The following illustration shows a data access page where the outermost group level that has two grouping fields.



Regardless of which technique you use to create a grouped page, you can set **GroupOn** and **GroupInterval** properties for each grouping field. You can also set the **DefaultSort** property to sort on multiple fields at each group level.

[On a datasheet or form in PivotTable or PivotChart view](#)

[Grouping on intervals in PivotTable and PivotChart view](#)

You can group [items](#) by specifying the type of interval and the size of the interval. For example, you can group a field with date values into different months, and specify the interval as 2, to create groups such as Jan-Feb, Mar-Apr, and so on.

The type of interval you can specify for a field depends on the data type of the field.

- **Numeric fields** You can group in intervals of any integer, such as 1, 2, 5, 100, and so on. For example, you can group the EmployeeID field into groups of 1-100, 101-200, and so on.
- **Date and time fields** You can group in intervals such as days, weeks, and quarters. For example, you can group the values in the ShippedDate field into months to show data for orders shipped in January, orders shipped in February, and so on.
- **Text fields** You can group based on the first *n* characters of the individual items. For example, you can group the LastName field by first letter to create groups, such as A, B, and so on.

You can specify the start and end values for the grouping range. For example, if you specify the start range as 01-Jul-1999 while grouping the ShippedDate field in weekly intervals, the following groups will be created:

<01-Jul-1999, 01-Jul-1999, 08-Jul-1999, ...

Dates prior to 01-Jul-1999 are grouped into a single group titled <01-Jul-1999.

If you also specify the end value as 31-Dec-1999, the following groups will be created:

<01-Jul-1999, 01-Jul-1999, 08-Jul-1999, ... 31-Dec-1999, >31-Dec-1999

Dates that fall after 31-Dec-1999 are grouped into a single group titled >31-Dec-1999.



[Custom grouping in PivotTable view](#)

You can randomly select items from a [row](#) or [column field](#) and group them into higher-level groups. For example, you can select from the Promotions row field all the promotions that run for a specific period and create a group. This would add a new row field named Promotions 2 above the Promotions row field with two members: Group1 and Other. You can change the caption of Group1 to Fixed and Promotions 2 to Category in the **Properties** dialog box.

The Other group will contain all items that you did not include in the Fixed

custom group. You can then select all the popular promotions from the Other group and create a new custom group that will be captioned Popular. After you do this, the Category field will have three members: Fixed, Popular, and Other.

The following illustration shows how the PivotTable view will look after the captions of the [custom group field](#) and [custom groups](#) have been changed.

		Qtr	
		+	-
Cate...	Promo...	Sal...	Co...
Fixed			
	Total		
Popular			
	Total		
Other			
	Total		

- 1 A custom group field. Contains custom groups as its items and appears as the parent of the field whose items you grouped.
- 2 Custom groups. Appear as the parents of items you explicitly selected to create the groups.
- 3 The Other group is the parent of the items you did not assign to any specific custom group.

[Creating and deleting custom groups and custom group fields](#)

- You cannot group items that belong to different fields or items that belong to parent items into a custom group.
- When you create the first custom group for a field, a custom group field is automatically added as the field's parent. Similarly, when you delete the last custom group for a field, the custom group field is automatically deleted.
- When you delete a custom group that is not the last custom group in the

field, the deleted group's members automatically move to the Other group.

- The groups you create do not affect source data.

[Moving a field whose parent is a custom group field](#)

When you move a field that has custom groups between row and column areas, the custom group fields that are based on the field move with the field.

If you move the field to the [filter area](#), the custom group fields are hidden. However, the bound field will show the custom group hierarchy in the drop-down list, so you can filter data by selecting custom groups or individual values. When you move the field back to a row or column area, the custom group fields will appear again.

You cannot move a field with custom groups to the detail area.

[Adding and removing nested custom groups](#)

You can select two or more custom groups to create a higher-level grouping. For example, you can group the members of the Category field into two groups: high priority promotions and low priority promotions.

The following illustration shows how the row area will look with nested custom groups.

1

Priority	Category	Promo...
High	Fixed	
		Total
	Popular	
		Total
Total		
Low	Other	
	Total	
Grand Total		

1 A custom group field that appears as the parent of the Category custom group field.

You can also remove a lower-level custom group. The following illustration shows what the data will look like after the Category field has been removed.

Priority	Promotions
High	
	Total
Low	
	Total
Grand Total	

[Show All](#)

Group records

[On a report](#)

1. Open the report in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And Grouping** box.
3. In the first row of the **Field/Expression** column, select the field or type the [expression](#) that you want to group records on.

The field or expression in the first row is the first grouping level (the largest set). The second row is the second grouping level, and so on.

Note When you fill in the **Field/Expression** column, Microsoft Access sets the **Sort Order** to **Ascending**. To change the sort order, select **Descending** from the **Sort Order** list.

4. Set **GroupHeader** or **GroupFooter** or both to **Yes** to create the [group level](#).
5. Set one or more of the following group properties:

GroupOn Specifies how you want the values grouped. The options you see depend on the [data type](#) of the field on which you're grouping. If you group on an expression, you see all the options for all data types.

GroupInterval Specifies any interval that is valid for the values in the field or expression you're grouping on.

KeepTogether Specifies whether Microsoft Access prints all or only part of a group on the same page.

Note The Employee Sales by Country report in the Northwind sample database has two group levels. To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Employee Sales by Country report in Design view.

[On a data access page](#)

Grouping records on a data access page is similar to grouping records on a report: You can create a hierarchy that groups records from general categories to specific details.

[Group records by each value in a grouping field. For example, group sales by employee name.](#)

1. Open the data access page in [Design view](#).
2. Select the control on which you want to group records, and do one of the following:
 - To group on one field from a table or query, click **Promote**  on the toolbar.
 - To group on all fields in a table or query, click **Group by Table**  on the toolbar.

Microsoft Access adds a group header that contains an [expand control](#) and a [record navigation section](#) that contains a [record navigation control](#). If you grouped on one field, Access moves the control you grouped on to the group header. If you grouped on a table or [query](#), Access moves all the controls—including controls bound to fields that are in that table or query and controls bound to [Lookup fields](#)—to the group header.

[Group records by using an expression.](#)

1. Open the data access page in [Design view](#).
2. If the expression will reference field names, make sure those fields have been added to the page.

Note Any field you reference in an expression must be added to the [data definition](#) of the page; the field is automatically added to the data definition when you create a control bound to that field. If you don't want to display the values in those controls in [Page view](#) or in Microsoft Internet Explorer, you can either hide or delete the controls. Even if you delete a control bound to a field used in an expression on the page, it won't be deleted from the page's data definition.

3. Add a [bound span control](#) for the expression by clicking **Bound Span**  on the toolbox and then clicking in the section where you want the control.
4. Select the control, and then click **Properties**  on the toolbar.

5. In the **ControlSource** property box, type an alias for the expression followed by a colon and the expression you want. In this example, FullName is the alias for the expression.

FullName: FirstName & " " & LastName

6. With the bound span control still selected, click **Promote**  on the toolbar.
7. In the property sheet of the control, set **TotalType** to **dscNone**.

Microsoft Access adds a group header that contains an [expand control](#), and a [record navigation section](#) that contains a [record navigation control](#). Access moves the control on which you grouped to the group header.

 [Group records on intervals of autonumbered, currency, or numeric values. For example, group sales in intervals of 1,000.](#)

1. Open the data access page in [Design view](#).
2. Make sure there are two controls bound to the field you want to use to group by interval. You'll use one to group on and the other to display the actual value.
3. Select one of the two controls and click **Promote**  on the toolbar.

Microsoft Access adds a group header that contains an [expand control](#), and a [record navigation section](#) that contains a [record navigation control](#). Access moves the control on which you grouped to the group header.

4. Right-click the [section bar](#) of the new group header, and then click **Group Level Properties**.
5. Set the following properties.
 - Set the **GroupOn** property to **dscInterval**.
 - Set the **GroupInterval** property to any number that's valid for the values in the field you grouped on.

 [Group records on intervals of date or time values. For example, group sales by month.](#)

1. Open the data access page in [Design view](#).
2. Make sure there are two controls bound to the field you want to use to group by interval. You'll use one to group on and the other to display the

actual value.

3. Select one of the two controls, and click **Promote**  on the toolbar.

Microsoft Access adds a group header that contains an [expand control](#), and a [record navigation section](#) that contains a [record navigation control](#). Access moves the control on which you grouped to the group header.

4. Right-click the [section bar](#) of the new group header, and then click **Group Level Properties**.
5. Set the following properties.
 - Set the **GroupOn** property to a value in the list. You can group records on values that fall in the same calendar year, calendar quarter, month, week, day, hour, or minute.
 - Set the **GroupInterval** property to any number that is valid for the values in the field you're grouping on. For example, if you are grouping on months, valid values for the **GroupInterval** property are **1** (a single month), **2** (a two-month span, such as January - February), **3** (a three-month span), **4** (a four-month span), or **6** (a six-month span).
6. If you want to display the value in the grouped field as an interval, do the following:

1. Add a [bound span control](#) for the expression by clicking **Bound Span**  in the toolbox and then clicking in the group header where the control you grouped on an interval is located.
2. Select the control, and then click **Properties**  on the toolbar.
3. Type the alias for the expression followed by a colon and the expression you want in the **ControlSource** property box. For example, if you grouped on ShippedDate and set the **GroupOn** property to **dscMonth** and the **GroupInterval** property to **1**, type the following expression to display the values in the GroupOfShippedDate field as a three-letter month:

Month: Format(GroupOfShippedDate, "mmm")

7. If you don't want the grouped control to be visible, double-click the control and set its **Visibility** property to **Hidden**.

[Group records by the first n characters in a text value. For example, group](#)

[products by the first letter of the product name.](#)

1. Open the data access page in [Design view](#).
2. Make sure there are two controls bound to the field you want to use to group by interval. You'll use one to group on and the other to display the actual value.
3. Select one of the two controls and click **Promote**  on the toolbar.

Microsoft Access adds a group header that contains an [expand control](#), and a [record navigation section](#) that contains a [record navigation control](#). Access moves the control on which you grouped to the group header.

4. Right-click the [section bar](#) of the new group header, and then click **Group Level Properties**.
5. Set the following properties.
 - Set the **GroupOn** property to **dscPrefix**.
 - Set the **GroupInterval** property to the number of characters you want to group on.

 [Group records on multiple fields.](#)

1. Open the data access page in [Design view](#).
2. Select the control that you want to use to create the outermost [group level](#), and click **Promote**  on the toolbar.
3. Repeat step 2 for the grouping control for the next highest group level and so on, until you finish creating all the levels you want.
4. Set the properties for each group level.

Notes

- If you add other controls to the [group header](#), Microsoft Access groups on every control in the group header. For example, if you add a control bound to the ShipRegion field to the Orders-ShipCountry header, Access uses the values in both the ShipCountry and ShipRegion controls to group the records.
- Use the **Group by Table**  button when the page contains fields from two

tables or [queries](#) that have a [one-to-many relationship](#) and you want to group on the table or query on the one side of the relationship. For example, in a page that contains fields from the Orders and Order Details tables, group on the Orders table. When you use **Group By Table** to group records, Access groups and sorts records by using the table's primary key — in this case, OrderID.

- When you group on a control, Access creates an alias for the control by adding the prefix "GroupOf" to the control's name in the **ControlSource** property; for example, GroupOfShippedDate: ShippedDate. When you refer to this control in an [expression](#) or in a property, you must use the alias.
- Set the **DefaultSort** group level property to specify a default sort order for each [group level](#).

[On a datasheet or form in PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Group items at intervals](#)

1. Select the row or column field whose [items](#) you want to group.
2. On the **PivotTable** toolbar, click **Properties** , and then click the **Filter and Group** tab.
3. Set **Group items by** to a value other than **(No grouping)**.
4. Set **Interval** to the value you want. For example, if you are grouping items by prefix characters, setting **Interval** to **1** would group items on A, B, C, and so on.
5. To specify the value at which grouping begins, select the **Start at** check box and then enter a value in the text box.
6. To specify the value at which grouping ends, select the **End at** check box and then enter a value in the text box.

Note The end value must be higher than the start value.

[Create a custom group](#)

1. Click the label of the parent field that contains the items you want to group, and then click **Expand**  on the **PivotTable** toolbar.

2. Click the first item that you want to include in the group.

To select more items, hold down the CTRL key and click each item.

3. Right-click a selected item, and then click **Group Items**.
4. Repeat steps 2 and 3 to create more groups.

Notes

- When you add the first [custom group](#) to a field, a [custom group field](#) is added to the field set. The custom group field will be the parent of the field whose items you grouped.
- You can change the caption of the custom group field and the custom groups in the **Properties** dialog box.

[On a datasheet or form in PivotChart view](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Select the [category field](#) or [series field](#) whose items you want to group.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Filter and Group** tab.
4. Set **Group items by** to a value other than (**No grouping**).
5. Set **Interval** to the value you want. For example, if you are grouping alphabetical items by prefix characters, setting **Interval** to **2** would group items based on the first two prefix characters.
6. To specify the value at which grouping begins or ends for numeric intervals, select either or both of the **Start at** or **End at** check boxes, and then enter a value in the adjacent text box.

[Show All](#)

Promote or demote a field, table, or query

Promoting a field creates a new [group level](#) immediately above the group level to which the field belongs. Demoting a field removes the group level to which the field belongs, and moves the field to the next lower group level. You can also promote all fields that belong to a table or [query](#).

1. Open the data access page in [Design view](#).
2. Do one of the following:

To promote a field, select the field and then click **Promote**  on the toolbar.

To promote a table or query, select a field that belongs to the table or query, and then click **Group by Table**  on the toolbar.

To demote a field, select the field and then click **Demote**  on the toolbar.

Note When you demote a field, all other fields that belong to the group level get demoted as well.

 **Tip**

When you click **Promote** , the selected control is moved to the next higher group level. To copy the field instead of moving it, press and hold the CTRL key when you click **Promote** on the toolbar.

[Show All](#)

Merge groups

When you merge two groups on a data access page, Microsoft Access moves the [controls](#) in the [group header](#) and [group footer](#) of the selected [group level](#) into the header and footer of the next lower group level.

1. Open the data access page in [Design view](#).
2. Click the [section bar](#) of the group header or footer for the group level you want to merge into the next lower group level.
3. Click **Demote**  on the toolbar.

[Show All](#)

Set the number of records to display in a group

1. Open the data access page in [Design view](#).
2. Scroll to a section that belongs to the group for which you want to specify the setting.
3. Right-click the [section bar](#), and then click **Group Level Properties**.
4. In the **DataPageSize** property box, select a value from the list or type a value in the box.

Notes

- If a [group level](#) doesn't have a [record navigation section](#) or a [record navigation control](#), you must set the **DataPageSize** property to **All** in order to see all the records for that group level. If that group level is the only group level on the page or the highest group level on the page, and if a page contains many records, it might take longer to load the page.
- You cannot set the **DataPageSize** property of a group level to a value other than **1**, if any of its sections include a PivotTable list, spreadsheet, or chart.

[Show All](#)

Keep a group of records together on a page

1. Open the report in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And Grouping** box.
3. Click the field or expression for the group level that you want to keep together.
4. In the group properties section of the **Sorting And Grouping** box, set the **KeepTogether** property to one of the following settings:

Whole Group The [group header](#), all detail records, and the [group footer](#) are printed on the same page.

With First Detail The group header is printed on a page only if the first detail record also fits on the same page.

5. Double-click the report selector.
6. Set the **GrpKeepTogether** property to **Per Page**.

Notes

- If you want to keep the data in one section together, set the **KeepTogether** property for that section to **Yes**.
- The Alphabetical List of Products report in the Northwind sample database keeps the products for each letter together as a group. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Alphabetical List of Products report in Design view.

[Show All](#)

Change sorting and grouping levels

[On a report](#)

1. Open the report in [Design view](#).
2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And Grouping** box.
3. Do one or more of the following:

[Change sorting or grouping order](#)

1. In the **Sorting And Grouping** box, click the [row selector](#) of the field or [expression](#) you want to move.
2. Click the selector again, and drag the row to a new location in the list.

Note If the groups you're rearranging have [headers](#) or [footers](#), Microsoft Access moves the headers, footers, and all the [controls](#) in them to their new positions. However, you must adjust the locations of the controls in the headers and footers yourself.

[Insert a sorting or grouping level](#)

1. In the **Sorting And Grouping** box, click the [row selector](#) of the row above which you want to insert the new field or [expression](#), and then press INSERT.
2. In the **Field/Expression** column of the blank row, select the field you want to sort on, or type an expression.

When you fill in the **Field/Expression** column, Microsoft Access sets the **Sort Order** to **Ascending**. To change the sort order, select **Descending** from the **Sort Order** list.

[Delete a sorting or grouping level](#)

1. In the **Sorting And Grouping** box, click the [row selector](#) of the field or [expression](#) you want to delete, and then press DELETE.

2. Click **Yes** to delete the sorting or grouping level, or click **No** to keep it.

Note If the field or expression you delete has a header or footer, Microsoft Access deletes the header or footer and its [controls](#).

[Modify the inherited sort order in a report](#)

1. Click **Properties**  on the toolbar to display the report's property sheet.
2. Click the **OrderBy** property box, and then specify the field and sort order you want to sort on.
 - To sort records in one field in ascending order, type the field name enclosed in brackets ([]) followed by ASC.

[fieldname1] **ASC**

- To sort records in one field in descending order, type the field name enclosed in brackets ([]) followed by DESC.

[fieldname1] **DESC**

- To sort records in more than one field in ascending or descending order, type the setting this way:

[fieldname1] **ASC**,*[fieldname2]* **DESC**

Note If you don't specify ASC or DESC after field names, Microsoft Access sorts in ascending order, the first listed field first, then the next listed field, and so on.

[Change group levels on a data access page](#)

1. Open the data access page in [Design view](#).
2. Do one or more of the following:

[Add a group level](#)

1. Click the [control](#) you want to move to a higher [group level](#).
2. Do one of the following:

- To group on one field from a table or [query](#), click **Promote**  on the toolbar.
- To group on all fields in a table or query, click **Group by Table**  on the toolbar.

Microsoft Access adds a [group header](#) that contains an [expand control](#) and a [record navigation section](#) that contains a [record navigation control](#). If you grouped on one field, Access moves the control on which you grouped to the group header. If you grouped on a table or query, Access moves all the controls—including controls bound to fields that are in that table or query and controls bound to [Lookup fields](#)—to the group header.

[Remove a group level](#)

1. Click the [section bar](#) of the [group header](#) that you want to move to a lower [group level](#).
2. Click **Demote**  on the toolbar.

Microsoft Access moves the controls in the selected group header and [footer](#) into the header and footer of the next lower group level.

Note To change a field that is used to group records, follow the steps to remove the group level, and then follow the steps to add a group level with the field you want.

[Show All](#)

Examples of grouped reports

 [Example of a report that groups records on text values](#)



- 1 Each group starts with the first letter of the product names in the group.
- 2 Within each group, the product names are listed alphabetically.

The following table lists the sorting and grouping specifications for this report. Product Name in the first row of the **Field/Expression** column in the **Sorting And Grouping** box (the second column in the table) groups the records by the first letter of the product names. Product Name in the second row (the third column in the table) sorts the records alphabetically by product name under each letter.

Field/ Expression	Product Name	Product Name
Sort Order	Ascending	Ascending
Group Header	Yes	No
Group Footer	Yes	No
GroupOn	Prefix Characters	Each Value
GroupInterval	1	1
KeepTogether	Whole Group	No

Notes

- To print the first letter of each group in the group header, you use the **Left** function.

- You can look at the Alphabetical List of Products report in the Northwind sample database. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Alphabetical List of Products report in [Design view](#).

 [Example of a report that groups records on currency values \(or AutoNumber or numeric values\)](#)



Sale Amount	
1	\$18
	\$40
	\$97
2	\$159
	\$189
	\$191

- Sales from \$0 to \$99
- Sales from \$100 to \$199

The following table lists the sorting and grouping specifications for this report. Sale Amount in the first row of the **Field/Expression** column in the **Sorting And Grouping** box (the second column in the table) groups the records in intervals of 100. Sale Amount in the second row (the third column) sorts the records in ascending amounts within each \$100 interval.

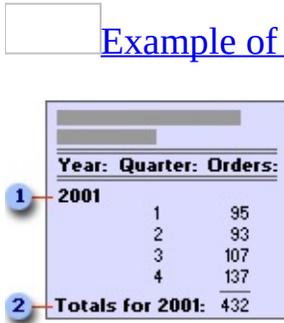
Field/ Expression	Sale Amount	Sale Amount
Sort Order	Ascending	Ascending
Group Header	No	No
Group Footer	Yes	No
GroupOn	Interval	Each Value
GroupInterval	100	1
KeepTogether	Whole Group	No

Notes

- Microsoft Access starts AutoNumber, Currency, and Number groups with 0 (zero). For example, if you set the **GroupOn** property to **Interval**, and set

the **GroupInterval** property to 5, Microsoft Access groups the records like this: 0 to 4, 5 to 9, 10 to 14, and so on.

- The Sales Totals by Amount report in the Northwind sample database groups records in intervals of 1,000. To view this report, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open the Sales Totals by Amount report in [Design view](#).

 [Example of a report that groups records on date \(or time\) values](#)



Year:	Quarter:	Orders:
2001	1	95
	2	93
	3	107
	4	137
Totals for 2001:		432

1 [Group header](#) showing year

2 [Group footer](#) showing totals for year

The following table lists the main sorting and grouping specifications for this report. Shipped Date in the first row of the **Field/Expression** column in the **Sorting And Grouping** box (the second column of the table) groups the records by year. Shipped Date in the second row (the third row of the table) groups the records for each year by quarter.

Field/ Expression	Shipped Date	Shipped Date
Sort Order	Ascending	Ascending
Group Header	Yes	No
Group Footer	Yes	Yes
GroupOn	Year	Qtr
GroupInterval	1	1
KeepTogether	Whole Group	No

Notes

- To print the year or quarter instead of a specific date, use the **DatePart** or **Format** function. To count the number of orders shipped, use the **Count** function.
- If you set the **Group On** property to **Hour** or **Minute** in the **Sorting and Grouping** dialog box, Microsoft Access groups records on unique hours or minutes starting from 1/1/1900. So, 10:00 a.m. on 1/1/1900 is different from 10:00 a.m. on 1/2/1900, and so on. To group records on hours or minutes, regardless of date, use the **Hour** or **Minute** function. For example, to see all flights that are scheduled to arrive at 10:00 a.m. on any day, group records on the following expression: =Hour([ArrivalTime])
- The Summary of Sales by Year report in the Northwind sample database prints quarterly sales for each year. The Summary of Sales by Quarter report in the Northwind sample database shows sales from multiple years for each quarter. To view one of these reports, open the Northwind database in the Samples subfolder of your Microsoft Office folder, and then open that report in [Design view](#).

[Show All](#)

About hyperlinks

[Hyperlinks](#) provide a useful and convenient way to navigate between objects.

[What is a hyperlink?](#)

A hyperlink is a pointer from one object to another. The destination is frequently another Web page, but it can also be a picture, an e-mail address, a file (such as a multimedia file or Microsoft Office document), or a program. The hyperlink itself can be displayed as text or as a picture.

You can use hyperlinks to do the following:

- Navigate to a file or Web page on a network, [intranet](#), or [Internet](#)
- Send an e-mail message
- Start a file transfer, such as downloading or an [FTP](#) process

When you point to text or a picture that contains a hyperlink, the pointer becomes a hand , indicating that it is something you can click on.

When a site visitor clicks the hyperlink, the destination is displayed, opened, or run, depending on the type of destination. For example, a hyperlink to a sound file opens the file in a media player, and a hyperlink to a Web page displays the page in the Web browser.

[Parts of a hyperlink address](#)

A [hyperlink address](#) can have up to four parts that are separated by the number sign (#):

displaytext#address#subaddress#screentip

Part	Description
<i>displaytext</i>	The text that you see in the field or control . For example, you might want to display a descriptive name for the World Wide Web site or

object specified by the *address* and the *subaddress*. This is an optional field.

Address An absolute or relative path to a target document. An absolute path is a fully qualified [URL](#) or [UNC](#) path to a document. For [data access pages](#), a relative path is a path that is related to the saved location of the page. For other database objects, a relative path is a path that is related to the **Hyperlink Base** setting in the **DatabaseName Properties** dialog box or to the current database path. This is an optional field as long as *subaddress* points to an object in the current [.mdb](#) or [.adp](#) file.

subaddress The location in the file or page. For example, you might want to point to a particular form or report in a [Microsoft Access database](#), or to a particular bookmark in a Microsoft Word document. This is an optional field.

screentip The text that appears when you rest the pointer on a hyperlink. This is an optional field.

You can type a hyperlink address directly in a [hyperlink field](#) in a datasheet or in a text box or control that accepts hyperlinks on a form or data access page.

Note In a field or text box that formats the hyperlink address as a hyperlink, Microsoft Access doesn't display all four parts in the field or control. If you want to see the whole hyperlink address, select the insertion point and press F2.

- If you enter *displaytext*, Access doesn't show any of the rest of the address following the display text.
- If you don't enter *displaytext*, Access displays just the *address*.
- *Subaddress* is displayed only if there is no *displaytext* or *address*.
- The [ScreenTip](#) that is displayed for a hyperlink on a data access page is defined for the hyperlink control, so a ScreenTip that you enter as part of the address won't be displayed on the data access page.

 [About the hyperlink control](#)

Use the hyperlink control when each record in the underlying [record source](#) has a different hyperlink address. You bind the hyperlink control to a Text field that contains hyperlink values, such as e-mail addresses.

As you move from record to record in [Page view](#) or Microsoft Internet Explorer,

you can click the hyperlink to either go to a different Web page or open a mail program to send an e-mail message.

[Image hyperlink control](#)

Use the image hyperlink control to add an image to a data access page that, when clicked, displays another Web page from your hard drive, the Web, or another location. When you create an image hyperlink control, you select the image to display on the page and the address of the file to jump to. In Page view, as the pointer passes over the image, the pointer changes to a hand, indicating that the image is a link that you can click to go to another page. You can also define **ScreenTips** and alternate text for the image.

The image hyperlink control is useful for providing links to information that relates to data on your data access page

[What is a URL and how does it work?](#)

When you create a hyperlink, its destination is encoded as a Uniform Resource Locator ([URL](#)), such as `http://example.microsoft.com/news.htm` or `file://ComputerName/SharedFolder/FileName.htm`. A URL contains a Web server or network location path and file name. A URL also identifies the protocol that will handle the file, such as HTTP, FTP, or FILE.

What is the difference between an absolute and a relative hyperlink?

An absolute [URL](#) contains the full destination address. A relative URL has one or more missing address parts. To make the address usable, the missing information is taken from the page that contains the URL. For example, if the protocol and domain are missing, the Web browser uses the protocol and domain of the current page. It is common for pages in a Web site to use relative URLs containing only a partial path and file name. If the files are moved to another server, any hyperlinks will continue to work as long as the relative positions of the pages remain unchanged. For example, a hyperlink on `Products.htm` points to a page in the same folder; if both pages are moved to the same folder on a different server, the URL in the hyperlink will still be correct.

When you create hyperlinks in Microsoft FrontPage, you do not need to know the destination URL—you can simply browse to the page or file in a local web,

network file system, or on the World Wide Web, and then FrontPage will provide the correct URL. When you create a hyperlink to a destination in the same web, FrontPage will create a relative URL. If you rename or move a file in a web, FrontPage will update all the URLs that point to the file. By generating relative URLs, FrontPage saves you time and ensures that your Web site will be portable.

[Examples of hyperlink addresses](#)

Note A hyperlink address can have up to four parts that are separated by the number sign (#): *displaytext#address#subaddress#screentip*

The following examples display the hyperlink address and what they point to. Note that [URLs](#) use slash marks and [UNC](#) paths use backslashes.

Microsoft Corporation#http://www.microsoft.com#

The Microsoft home page on the Web. Displayed as "Microsoft Corporation."

#http://www.microsoft.com#

The Microsoft home page on the Web.

#http://www.microsoft.com#sample#

The **Name** tag called "sample" in the .html document for the Microsoft home page. **Name** is an attribute of some tags in HTML pages.

#\\northwind\public\products.doc##List of discounted products

A Microsoft Word file called Products.doc in the public folder on the Northwind server. The ScreenTip is "List of discounted products."

Resume#c:\windows\personal\resume.doc#

A Word file called Resume.doc located in the C:\Windows\Personal folder. Displayed as "Resume."

#c:\windows\personal\resume.doc#

The same Resume.doc file located in the C:\Windows\Personal folder. The path

name will be displayed because there's no display text.

#c:\windows\personal\resume.doc#Qualifications

The bookmark called "Qualifications" in the Resume.doc file.

#c:\windows\personal\ northwind.mdb#Form Suppliers

The Suppliers form in the Northwind database, located in the C:\Windows\Personal folder.

Suppliers##Form Suppliers#Click to view and edit supplier information

The Suppliers form in the current database. Displayed as "Suppliers," with the ScreenTip "Click to view and edit supplier information."

#c:\windows\personal\1996 Sales.ppt#13

Slide 13 in the 1996 Sales.ppt PowerPoint presentation in the C:\Windows\Personal folder.

#c:\windows\personal\budget. xls#Sheet1!A2

Cell A2 in Sheet1 of the budget.xls Excel file, located in the C:\Windows\Personal folder.

To make it easier to enter a hyperlink address, Access allows you to omit certain parts of an address. For example:

- If you type **http://example.microsoft.com**, Access recognizes the "http:" protocol and translates the address to #http://example.microsoft.com#. (You don't have to enter the number signs (#) yourself.)
- If the text that you enter doesn't include a number sign (#), a protocol (such as "http:"), or a reference to an object in the current database, Access assumes the text entered is display text and appends "http://" to the same text for the *address*. For example, if you type **example.microsoft.com**, Access translates the address as example.microsoft.com#http://example.microsoft.com#.
- If you want to point to an Access object in the current database, you can specify just the object name that you want to link to— you don't have to

specify a UNC path. If there are multiple objects with the same name, Access uses the following order to determine which one to reference: forms, reports, queries, tables, data access pages, macros, modules, views, schemas, stored procedures, SQL linked tables, linked tables, and triggers. If, for example, there is both an Orders form and an Orders table, Access assumes that the reference is to the Orders form. You can specify a particular object type by entering *object type object name*. So to refer to the Orders table instead of the Orders form, enter **table Orders** as the *subaddress*. For object types with multiple words, omit the spaces between words; for example, enter **dataaccesspage Orders**.

 [Some hyperlink subaddress syntax examples in Office](#)

**To go to
location in
this type
of file**

Enter

The object type and object name. For example, enter **table Customers** to go to the Customers table.

Notes

Microsoft
Access

On a non-English system, Access recognizes the object type in English and in the local language; however, hyperlinks created by using the **Insert Hyperlink** dialog box all have the English-language string for an object type.

Microsoft
Excel

A defined name or a *sheet!range*. For example, enter **Sheet1!A2** to go to the A2 cell in Sheet1.

Microsoft
PowerPoint

A slide number

Microsoft
Word

A bookmark name

HTML
document

A bookmark name (defined by using the NAME attribute of the <A> tag)

[Show All](#)

Change a hyperlink

You can change either the text that is displayed or the address that the [hyperlink](#) points to.



[Modify the hyperlink of a label, picture, or command button](#)

1. Open the form or report in Design view.
2. Right-click the label, [image control](#), or command button to display the shortcut menu.
3. On the shortcut menu, point to **Hyperlink**, and then click **Edit Hyperlink**.
4. In the **Edit Hyperlink** dialog box, click an option under **Link to**, and then fill in the information required for that option.
5. In the **Text to display** box, type the text you want displayed in the control.
6. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access displays the hyperlink address, the name of the database object, or the e-mail address as the ScreenTip.

Note You can also modify the hyperlink by typing directly in the [property sheet](#).



[Modify a hyperlink that's stored in a table or displayed in a text box or combo box](#)

1. Open the table, query, or form in [Datasheet view](#).
2. Right-click the [hyperlink field](#) in the datasheet. In the form, right-click the text box or combo box that contains the hyperlink.
3. To add or modify the display text, point to **Hyperlink** on the shortcut menu. In the **Display Text** box, type the display text you want.
4. To change a hyperlink address, point to **Hyperlink** on the shortcut menu, and then click **Edit Hyperlink**.
5. In the **Edit Hyperlink** dialog box, click an option under **Link to**, and then fill in the information required for that option.

Change the default formatting of hyperlinks

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab.
3. Click **Web options**.
4. Select the options you want.

[Show All](#)

Create a field to store hyperlinks (MDB)

1. Open an existing table in [Design view](#), or create a new table.
2. In the **Field Name** portion of the window, type a name for the field that you'll use to store the hyperlinks.
3. In the **Data Type** column for that field, select **Hyperlink**.
4. Click **Save**  on the [toolbar](#) to save the table.

Notes

- In [Datasheet view](#), you can create a field to store hyperlinks by clicking the **Hyperlink Column** command on the **Insert** menu and then typing the hyperlink address in the new field.
- If you create a table by importing data, Microsoft Access automatically converts any column that contains [URLs](#) or [UNC](#) paths into a [hyperlink field](#). Access will convert the column only if all the values start with a recognized protocol, such as "http:" or "\\". If any value starts with an unrecognized protocol, Access won't convert the column into a hyperlink field.

[Show All](#)

Create a hyperlink

When you create a [hyperlink](#) by using any of the following methods, Microsoft Access creates a [label control](#) on the form or report. Although hyperlinks in a report won't work when viewed in Access, the hyperlinks will work when you output the report to Microsoft Word, to Microsoft Excel, or to [HTML](#).

Notes

- You can open the target document or Web page by clicking the hyperlink in a table or query, or in a form in [Datasheet view](#) or [Form view](#).
- In a [data access page](#), the text box in which you enter a hyperlink address displays the hyperlink address as a text string, not as a formatted hyperlink. The formatted address can be found in the **Hyperlink** property of the text box.

The [ScreenTip](#) displayed for a hyperlink on a data access page is defined in the hyperlink control, so a ScreenTip that you enter won't be displayed on the page.

To see a newly created hyperlink address displayed as a formatted hyperlink on the data access page, click **Refresh** on the Microsoft Internet Explorer toolbar.

[Type a hyperlink in a field or text box](#)

1. Open the table, query, or form in Datasheet view, the form in Form view, or the data access page in [Page view](#).
2. Place the insertion point in the [hyperlink field](#) in the datasheet or in the text box or other control that accepts hyperlinks on the form or data access page.
3. Type a valid hyperlink address. A defined hyperlink address can have up to four parts separated by the number sign (#):

displaytext#address#subaddress#screentip

The following example includes *displaytext*, *address*, and *screentip*. Note that it also includes the # for *subaddress*, which has been omitted:

Microsoft Corporation#http://example.microsoft.com/##Example page for Microsoft Corporation

Access adds the hyperlink to the field or control.

[Paste a hyperlink from a Microsoft Office document](#)

1. Open an Office document that contains a hyperlink.
2. Right-click the hyperlink that you want to copy. Point to **Hyperlink** on the shortcut menu, and then click **Copy Hyperlink**.

Note When you copy a hyperlink in an Office document, make sure to copy just the hyperlink itself. If you copy characters that surround the hyperlink but aren't part of the hyperlink, Access will link to the text that you selected instead of to the hyperlink address.

3. Switch to Access and open a form or report in Design view.
4. On the **Edit** menu, click **Paste as Hyperlink**.

Access adds a label to the form or report. To test the link, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Link to text in a Microsoft Office document](#)

1. Open the Office document that you want to link to. For example, open a Microsoft Word document.

Note You can create a hyperlink by using text that you select in a data access page, but not text pasted from other Access objects.

2. Select the text you want to link to.
3. On the **Edit** menu, click **Copy**.
4. Switch to Access and open a form or report in Design view.
5. On the **Edit** menu, click **Paste as Hyperlink**.

Access adds a label to the form or report. To test the link, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Drag an Internet shortcut to a form or report](#)

1. Open a form in Design view or a report in Design view.
2. Locate the desired Internet shortcut (such as an icon on your desktop).
3. Select the Internet shortcut.
4. Use the right mouse button to drag the selection to the form or report.

Access adds a label to the form or report. To test the link, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Create a hyperlink between data access pages in the current database](#)

1. Open a data access page in Design view.
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **Page in this database** under **Link to**.
4. In the **Text to display** box, type the text that you want to be displayed in the field or text box. If you leave the box blank, Access will use the hyperlink address as the display text.
5. Click **ScreenTip** and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address as the ScreenTip.
6. From the list, select the data access page you want to link to.
7. To create a filter for the data access page, click **Server Filter**. In the **Server Filter** dialog box, type an expression in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** text box. For example, if you have a data access page that displays order information, including the ID of the salesperson, and you want to link to a second data access page with the employee record about that salesperson, type **EmployeeID=[EmployeeID]**.
8. Click **OK**.

Access adds the hyperlink to the data access page. To test the link, switch to Page view and click the hyperlink. Access opens the target page in your Web browser.

[Create a link to an existing file or Web page](#)

1. Open a form, a report, or a data access page in Design view.
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **Existing file or Web page** under **Link to**.

4. In the **Text to display** box, type the display text for the hyperlink. If you leave it blank, Access uses the hyperlink address and/or subaddress as the display text.
5. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address and/or subaddress as the ScreenTip.
6. In the **Address** box, type a path or a [URL](#).
7. Click **Current Folder**, **Browsed Pages**, or **Recent Files**, and then click an item from the list. If you click **Current Folder** and don't see the object you want, select a different folder from the **Look in** box.

Note You can go to a specific location in an [HTML](#) file or to an object in the current [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp). Click **Bookmark**, select the [bookmark](#) or object you want in the list, and then click **OK**.

8. To create a filter (data access page only), click **Server Filter**. In the **Server Filter** dialog box, type an expression in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** text box. For example, if you have a data access page that displays order information, including the ID of the salesperson, and you want to link to a second data access page with the employee record about that salesperson, type **EmployeeID=[EmployeeID]**.
9. Click **OK** in the **Insert Hyperlink** box.

On a data access page, Access adds the hyperlink in a [hyperlink control](#). In a form or report, it adds the hyperlink in a label. To test the link on a data access page, switch to Page view and click the hyperlink. To test the link on a form or report, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Create a link to a table, query, form, report, or data access page](#)

1. Open a form or a report in Design view.
2. Click **Insert Hyperlink**  on the toolbar.
3. In the **Insert Hyperlink** dialog box, click **Object in this database** under **Link to**.
4. In the **Text to display** box, type the text that you want displayed in the field or text box. If you leave the box blank, Access will use the type of object

and the object name as the display text.

5. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the type of object and the object name as the ScreenTip.
6. From the list, select the database object that you want to link to.
7. Click **OK**.

Access adds the hyperlink in a label. To test the link, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

Note To link to an object in another database, select the desired database file, and then click **Bookmark** to list the objects within that database.

 [Create a link that opens your e-mail program and creates a message addressed to the specified e-mail address](#)

1. Open a form, a report, or a data access page in Design view.
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **E-mail address** under **Link to**.
4. In the **Text to display** box, type the text that you want displayed in the field or text box. If you leave the box blank, Access will use the e-mail address as the display text.
5. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the e-mail address as the ScreenTip.
6. In the **E-mail address** box, type an e-mail address. You can also select an address from the **Recently used e-mail addresses** list.
7. In the **Subject** box, type the information that you want to appear in the Subject field of the e-mail message.

Note Some Web browsers and e-mail programs might not recognize the subject line.

8. Click **OK**.

On a data access page, Access adds the hyperlink in a [hyperlink control](#). In a form or report, it adds the hyperlink in a label. To test the link on a page, switch to Page view and click the hyperlink. To test the link in a form or

report, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**. Access opens your mail program and creates a new message, with the address and subject filled in.

 [Create a new data access page and create a link to the page at the same time](#)

1. Open a form, a report, or a data access page in Design view.
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **Create new page** under **Link to**.
4. In the **Text to display** box, type the text that you want to be displayed in the field or text box. If you leave the box blank, Access will use the hyperlink address and/or subaddress as the display text.
5. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address and/or subaddress as the ScreenTip.
6. In the **Name of new document** box, type a name for the new page.

 [Tip](#)

You can create a different type of document, for example, a Microsoft Word file (.doc) or a Microsoft Excel spreadsheet (.xls), by typing the three-letter file extension for the type of file you want to create. However, you can't create a [Microsoft Access database \(.mdb\)](#) or a [Microsoft Access project \(.adp\)](#).

7. To create a filter (data access page only), click **Server Filter**. In the **Server Filter** dialog box, type an expression in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** text box. For example, if you have a data access page that displays order information, including the ID of the salesperson, and you want to link to a second data access page with the employee record about that salesperson, type **EmployeeID=[EmployeeID]**.
8. Access creates the new page in the current folder. To specify a different location, click **Change**.
9. Select an option under **When to edit**. You can edit the new page immediately or wait until later.
10. Click **OK**.

Access adds the hyperlink in a [hyperlink control](#) to the data access page. In a form or report, it adds the hyperlink in a label. To test the link on a data access page, switch to Page view and click the hyperlink. To test the link on a form or report, right-click the label, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Show All](#)

Create a picture or command button that works as a hyperlink

You can use a picture as a [hyperlink](#) on a form, report, or data access page. Although hyperlinks in a report won't work when viewed in Microsoft Access, the hyperlinks will work when you output the report to Microsoft Word, to Microsoft Excel, or to [HTML](#).

By using a picture

 [Create a hyperlink to an existing file or Web page](#)

1. Open a form, a report, or a data access page in [Design view](#).
2. On a form or report, click the **Image** tool  in the [toolbox](#). On a data access page, click the **Image Hyperlink** tool .
3. Click where you want to create the picture.
4. In the **Insert Picture** dialog box, locate the picture that you want to use, and then click **OK**.

Access adds an [image control](#) to the form or report.

Note If you are creating an **Image Hyperlink** on a data access page, skip to step 6.

5. On the form or report, make sure that the image control is selected, and then click **Properties**  on the toolbar to open the property sheet for the control.
6. In the **HyperlinkAddress** property, click the **Build** button  to open the **Insert Hyperlink** dialog box.
7. In the **Insert Hyperlink** dialog box, click **Existing file or Web page** under **Link to**.
8. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address and/or subaddress as the ScreenTip.
9. In the **Address** box, type a path or a [URL](#).
10. Click **Current Folder**, **Browsed Pages**, or **Recent Files**, and then click an item from the list. If you click **Current Folder** and don't see the object you want, select a different folder from the **Look in** box.

Note To go to a specific location in an HTML file or to an object in the current [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), click **Bookmark**. Click the [bookmark](#) or object you want in the list, and then click

OK.

To test the link on a form or report, right-click the picture, point to **Hyperlink** on the shortcut menu, and click **Open**.

To test the link on a page, switch to [Page view](#) and click the picture.

 [Create a new data access page and a picture that's a hyperlink to that page](#)

1. Open a form, a report, or a data access page in [Design view](#).
2. On a form or report, click the **Image** tool  in the [toolbox](#). On a page, click the **Image Hyperlink** tool .
3. Click where you want to create the picture.
4. In the **Insert Picture** dialog box, locate the picture that you want to use, and then click **OK**.

Access adds an image control to the form or report.

Note If you are creating an **Image Hyperlink** on a data access page, skip to step 6.

5. On the form or report, make sure that the image control is selected, and then click **Properties**  on the toolbar to open the property sheet for the control.
6. In the **HyperlinkAddress** property, click the **Build** button  to open the **Insert Hyperlink** dialog box.
7. In the **Insert Hyperlink** dialog box, click **Create new page** under **Link to**.
8. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address and/or subaddress as the ScreenTip.
9. In the **Name of new document** box, type a name for the new data access page.

 [Tip](#)

You can create a different type of document, for example, a Word file (.doc) or an Excel spreadsheet (.xls), by typing the three-letter file extension for the type of file that you want to create. However, you can't create a

[Microsoft Access database \(.mdb\)](#) or a [Microsoft Access project \(.adp\)](#).

10. Access creates the new document in the current folder. To specify a different location, click **Change**.
11. Select an option under **When to edit**. You can edit the new document immediately or wait until later.

To test the link on a form or report, right-click the picture, point to **Hyperlink** on the shortcut menu, and then click **Open**.

To test the link on a data access page, switch to [Page view](#) and click the picture.

[Create a hyperlink between two data access pages](#)

1. Open a data access page in [Design view](#).
2. Click the **Image Hyperlink** tool  in the [toolbox](#), and then click where you want to create the picture.
3. In the **Insert Picture** dialog box, locate the picture that you want to use, and then click **OK**.
4. In the **Insert Hyperlink** dialog box, click **Page in this database** under **Link to**.
5. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address as the ScreenTip.
6. From the list, select the page that you want to link to.
7. To create a filter for the page, type an [expression](#) in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** box. For example, to open the second page, which displays the records for the employee ID that is shown on the first page, type **EmployeeID=[EmployeeID]**.

Access adds the hyperlink to the data access page. To test the link, switch to [Page view](#) and click the hyperlink. Access opens the target page in your Web browser.

[Create a hyperlink to an object in the current database](#)

1. Open a form or a report in [Design view](#).
2. Click the **Image** tool  in the [toolbox](#), and then click where you want to create the picture.

3. In the **Insert Picture** dialog box, locate the picture you want to use, and then click **OK**.

Access adds an image control to the form or report.

4. Make sure that the image control is selected, and then click **Properties**  on the toolbar to open the property sheet for the control.
5. In the **HyperlinkAddress** property, click the **Build** button  to open the **Insert Hyperlink** dialog box.
6. In the **Insert Hyperlink** dialog box, click **Object in this database** under **Link to**.
7. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the object name as the ScreenTip.
8. From the list, select the database object that you want to link to.

To test the link, right-click the picture, point to **Hyperlink** on the shortcut menu, and then click **Open**.

Note To link to an object in another database, select the desired database file and then click **Bookmark** to list the objects within that database.

 [Create a picture that creates an e-mail message when clicked](#)

1. Open a form, a report, or a data access page in [Design view](#).
2. On a form or report, click the **Image** tool  in the [toolbox](#). On a data access page, click the **Image Hyperlink** tool .
3. Click where you want to create the picture.
4. In the **Insert Picture** dialog box, locate the picture that you want to use, and then click **OK**.

Access adds an image control to the form or report.

Note If you are creating an **Image Hyperlink** on a data access page, skip to step 6.

5. On the form or report, make sure the image control is selected, and then click **Properties**  on the toolbar to open the property sheet for the control.

6. In the **HyperlinkAddress** property, click the **Build** button  to open the **Insert Hyperlink** dialog box.
7. In the **Insert Hyperlink** dialog box, click **E-mail address** under **Link to**.
8. Click **ScreenTip**, and type the text you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the e-mail address as the ScreenTip.
9. In the **E-mail address** box, type an e-mail address. You can also select an address from the **Recently used e-mail addresses** list.
10. In the **Subject** box, type the information that you want to appear in the Subject field of the e-mail message.

Note Some Web browsers and e-mail programs might not recognize the subject line.

To test the link in a form or report, right-click the picture, point to **Hyperlink** on the shortcut menu, and then click **Open**.

To test the link on a data access page, switch to [Page view](#) and click the picture. Access opens your mail program and creates a new message, with the address and subject filled in.

By using a command button

[Create a hyperlink to an existing file or Web page](#)

1. Open a form in [Design view](#).
2. If the **Control Wizards** tool  in the [toolbox](#) is [selected](#), click it to turn off the Control Wizard.
3. In the toolbox, click the **Command Button** tool .
4. On the form, click where you want to place the command button.
5. Make sure the command button is selected, and then click **Properties**  on the toolbar to open the property sheet for the command button.
6. In the **HyperlinkAddress** property, click the **Build** button .
7. In the **Insert Hyperlink** dialog box, click **Existing File or Web Page** under **Link to**.
8. In the **Text to display** box, type the display text for the hyperlink. Access automatically displays the value of the command button's **Caption** property in the box.
9. Click **ScreenTip**, and type the text you want to appear when the user rests the mouse pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access displays the hyperlink address (and subaddress) as the ScreenTip.
10. In the **Address** box, type a path or a [URL](#).
11. Click **Current Folder**, **Browsed Pages**, or **Recent Files**, and then click an item from the list. If you click **Current Folder** and don't see the object you want, select a different folder from the **Look in** box.

Note To go to a specific location in an HTML file or to an object in the current [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), click **Bookmark**. Click the [bookmark](#) or object you want in the list, and then click **OK**.

To test the link, right-click the command button, point to **Hyperlink** on the shortcut menu, and then click **Open**.

[Create a new data access page and a command button that's a hyperlink to](#)

[that page](#)

1. Open a form in [Design view](#).
2. If the **Control Wizards** tool  in the [toolbox](#) is [selected](#), click it to turn off the Control Wizard.
3. In the toolbox, click the **Command Button** tool .
4. On the form, click where you want to place the command button.
5. Make sure the command button is selected, and then click **Properties**  on the toolbar to open the property sheet for the command button.
6. In the **HyperlinkAddress** property, click the **Build** button .
7. In the **Insert Hyperlink** dialog box, click **Create New Page** under **Link to**.
8. In the **Text to display** box, type the display text for the hyperlink. Access automatically displays the value of the command button's **Caption** property in the box.
9. Click **ScreenTip**, and type the text you want to appear when the user rests the mouse pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access displays the hyperlink address as the ScreenTip.
10. In the **Name of new document** box, type a name for the page.

 [Tip](#)

You can create a different type of document, for example, a Word file (.doc) or an Excel spreadsheet (.xls), by typing the three-letter file extension for the type of file you want to create. However, you can't create a [Microsoft Access database](#) (.mdb) or a [Microsoft Access project](#) (.adp).

11. Access creates the new page in the current folder. To specify a different location, click **Change**.
12. Click an option under **When to edit**. You can edit the new page immediately or wait until later.

To test the link, right-click the command button, point to **Hyperlink** on the shortcut menu, and then click **Open**.

 [Create a hyperlink to an object in the current database](#)

1. Open a form in [Design view](#).
2. If the **Control Wizards** tool  in the [toolbox](#) is [selected](#), click it to turn off

the Control Wizard.

3. In the toolbox, click the **Command Button** tool .
4. On the form, click where you want to place the command button.
5. Make sure the command button is selected, and then click **Properties**  on the toolbar to open the property sheet for the command button.
6. In the **HyperlinkAddress** property, click the **Build** button .
7. In the **Insert Hyperlink** dialog box, click **Object in This Database** under **Link to**.
8. In the **Text to display** box, type the display text for the hyperlink. Access automatically displays the value of the command button's **Caption** property in the box.
9. Click **ScreenTip**, and type the text you want to appear when the user rests the mouse pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access displays the hyperlink subaddress as the ScreenTip.
10. Click the database object you want to go to from the list.

To test the link, right-click the command button, point to **Hyperlink** on the shortcut menu, and then click **Open**.

Note To link to an object in another database, select the desired database file and then click **Bookmark** to list the objects within that database.

 [Create a command button that creates an e-mail message when clicked](#)

1. Open a form in [Design view](#).
2. If the **Control Wizards** tool  in the [toolbox](#) is [selected](#), click it to turn off the Control Wizard.
3. In the toolbox, click the **Command Button** tool .
4. On the form, click where you want to place the command button.
5. Make sure the command button is selected, and then click **Properties**  on the toolbar to open the property sheet for the command button.
6. In the **HyperlinkAddress** property, click the **Build** button .
7. In the **Insert Hyperlink** dialog box, click **E-mail Address** under **Link to**.
8. In the **Text to display** box, type the display text for the hyperlink. Access automatically displays the value of the command button's **Caption** property in the box.
9. Click **ScreenTip** and type the text you want to appear when the user rests the mouse pointer on the hyperlink. If you leave the **ScreenTip text** box

blank, Access displays the e-mail address as the ScreenTip.

10. In the **E-mail address** box, type an e-mail address, or click an address in the **Recently used e-mail addresses** list.
11. In the **Subject** box, type the information you want to appear in the Subject field of the e-mail message.

Note Some Web browsers and e-mail programs might not recognize the subject line.

To test the link, right-click the command button, point to **Hyperlink** on the shortcut menu, and click **Open**. When you click the hyperlink, Access opens your mail program and creates a new message, with the address and subject filled in.

[Show All](#)

Delete a hyperlink

1. Open the table, query, or form in [Datasheet view](#).
2. Go to the record, picture, or control that has the [hyperlink](#) you want to delete.
3. In a datasheet, right-click the [hyperlink field](#). In the form, right-click the [text box](#) or [combo box](#) that contains the hyperlink.
4. On the shortcut menu, point to **Hyperlink**, and then click **Remove Hyperlink**.

Note You can delete all the hyperlinks stored in a table by deleting the field that stores hyperlinks from the table's design.

[Show All](#)

Enter a hyperlink into an existing hyperlink address

There are several ways to change the [destination](#) address of an existing hyperlink. The method that you choose will depend on the object that contains the hyperlink.

[Enter a hyperlink to an object in the current Access file](#)

1. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).
2. Place the insertion point in the hyperlink field in the datasheet, or in the text box or combo box, that accepts hyperlinks on the form.
3. Click **Insert Hyperlink**  on the **Design** toolbar.
4. In the **Insert Hyperlink** dialog box, click **Object in this database** under **Link to**.
5. In the **Text to display** box, type the text that you want displayed in the field or text box. If you leave the box blank, then Access will use the object type and object name as the display text.
6. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip** box blank, Access displays the object type and object name as the ScreenTip.
7. Select the database object that you want to link to from the list.

Access adds the hyperlink to the field or control. You can click the hyperlink to open the target object.

[Enter a hyperlink that links to text in a Microsoft Office document or data access page](#)

1. Open the data access page or Office document that you want to link to. For example, open a Microsoft Word document.

Note You cannot link to text in other Access objects, only to text in data

access pages.

2. Select the text that you want to link to.
3. On the **Edit** menu, click **Copy**.
4. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).
5. Place the insertion point in the hyperlink field in the datasheet, or in the text box or other control, that accepts hyperlinks on the form.
6. Click **Paste** on the **Edit** menu.

Access adds the hyperlink to the field or control. On a form or datasheet, you can click the hyperlink to open the target document or page.

 [Enter a hyperlink in a table, query, or form by pasting a hyperlink or hyperlink address](#)

1. Navigate to a hyperlink or hyperlink address in one of the following ways:
 - Open a Microsoft Office document that contains an existing hyperlink.
 - Open a document that was created with any application and contains a valid hyperlink address that is not defined as a hyperlink.
 - Open a Web browser such as Microsoft Internet Explorer.
2. Do one of the following:
 - In an Office document, right-click the defined hyperlink. Point to **Hyperlink** on the shortcut menu, and then click **Copy Hyperlink**.
 - In any document, select the hyperlink address, and then click **Copy** on the **Edit** menu.
 - In the Web browser, select the address for the current page or select a valid hyperlink address anywhere on a page, right-click, and then click **Copy** or **Copy Shortcut**.

Note If you're copying a defined hyperlink from an Office document, make sure to copy just the hyperlink. If you copy characters that surround but aren't part of the hyperlink, Access will link to the text that you selected instead of using the address of the hyperlink.

3. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).

4. Place the insertion point in the hyperlink field in the datasheet, or in the text box or other control, that accepts hyperlinks on the form.
5. Do one of the following:
 - To paste a defined hyperlink, click **Paste as Hyperlink** on the **Edit** menu.
 - To paste a hyperlink address that isn't defined as a hyperlink, click **Paste** on the **Edit** menu. When you paste an address this way, Access recognizes the protocol in the address and converts the text to a hyperlink.

Access adds the hyperlink to the field or control. You can click the hyperlink to open the target document or Web page.

 [Enter a hyperlink in a data access page by pasting a hyperlink or hyperlink address](#)

1. Navigate to a hyperlink or hyperlink address in one of the following ways:
 - Open a document that contains the hyperlink or the hyperlink address.
 - In a Web browser, such as Microsoft Internet Explorer, go to the page containing the hyperlink or hyperlink address.
2. Do one of the following:
 - In a document, select the defined hyperlink or the hyperlink address and then click **Copy** on the **Edit** menu.
 - In a Web browser, select the address for the current page or select a valid hyperlink address anywhere on a page, right-click the selection, and then click **Copy** or **Copy Shortcut** on the shortcut menu.
3. Open a data access page in your Web browser.
4. Place the insertion point in the text box that accepts hyperlinks on the page.
5. On the **Edit** menu, click **Paste**.

On a data access page, the text box in which you enter a hyperlink address doesn't format the address as a hyperlink.

 [Tip](#)

To see the hyperlink address displayed as a formatted hyperlink, refresh the page in your Web browser.

 [Enter a hyperlink to a new data access page in a datasheet or form](#)

1. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).
2. Place the insertion point in the hyperlink field in the datasheet, or in the text box or combo box, that accepts hyperlinks on the form.
3. Click **Insert Hyperlink**  on the **Design** toolbar.
4. In the **Insert Hyperlink** dialog box, click **Create new page** under **Link to**.
5. In the **Text to display** box, type the text that you want displayed in the field or text box. If you leave the box blank, Access will use the hyperlink address as the display text.
6. Click **ScreenTip** and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip** box blank, Access will display the hyperlink address as the ScreenTip.
7. In the **Name of new document** box, type a name and the three-letter file name extension of the type of file that you want to create. If you omit the extension or enter .htm or .html, Access will create a new data access page.
8. Access creates the new document in the current folder. To specify a different location, click **Change**.
9. Select an option under **When to edit**. You can edit the new document immediately or wait until later.

Access adds the hyperlink to the field or control. You can click the hyperlink to open the target document.

[Enter a hyperlink by dragging an Internet shortcut](#)

1. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), the form in Datasheet view or [Form view](#), or the data access page in [Page view](#).
2. Locate the desired Internet shortcut (such as an icon on your desktop).
3. Select the Internet shortcut.
4. Left-click the hyperlink and drag it to the hyperlink field in the datasheet or to the text box or other control that accepts hyperlinks on the form or data access page.

Access adds the hyperlink to the field or control. On a form or datasheet, you can click the hyperlink to open the destination document or Web page. On a data access page, the text box into which you enter a hyperlink

address doesn't format the address as a hyperlink.



To see the hyperlink address displayed as a formatted hyperlink, refresh the page in your Web browser.

[Enter a hyperlink that creates an e-mail message in Form view or Datasheet view](#)

1. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).
2. Place the insertion point in the hyperlink field in the datasheet, or in the text box or combo box, that accepts hyperlinks on the form.
3. Click **Insert Hyperlink**  on the **Design** toolbar.
4. In the **Insert Hyperlink** dialog box, click **E-mail Address** under **Link to**.
5. In the **Text to display** box, type the text that you want displayed in the field or text box. If you leave the box blank, Access will use the e-mail address and subject as the display text.
6. Click **ScreenTip**, and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip** box blank, Access will display the e-mail address and subject as the ScreenTip.
7. In the **E-mail address** box, type an e-mail address. You can also select an address from the **Recently used e-mail addresses** list.
8. In the **Subject** box, type the information that you want to appear in the Subject field of the e-mail message.

When you click the hyperlink, Microsoft Access opens your mail program and creates a new message, with the address and subject filled in.

Note Some Web browsers and e-mail programs might not recognize the subject line.

[Enter a hyperlink to an existing file or Web page](#)

1. In a [Microsoft Access database](#) (.mdb) or [Microsoft Access project](#) (.adp), open the table or query in [Datasheet view](#), or the form in Datasheet or [Form view](#).

2. Place the insertion point in the hyperlink field in the datasheet, or in the text box or combo box, that accepts hyperlinks on the form.
3. Click **Insert Hyperlink**  on the **Design** toolbar.
4. In the **Insert Hyperlink** dialog box, click **Existing file or Web page** under **Link to**.
5. In the **Text to display** box, type the display text for the hyperlink. If you leave the box blank, Access will use the hyperlink address as the display text.
6. Click **ScreenTip**, and then type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip** box blank, Access will display the hyperlink address as the ScreenTip.
7. Do one of the following:
 - In the **Address** box, type a path or a [URL](#).
 - Click **Current Folder**, **Browsed Pages** or **Recent Files** and then click an item from the list. If you click **Current Folder** and don't see the object you want, select a different folder from the **Look in** box.
8. To go to a specific location in an HTML file or to an object in the current Microsoft Access database (.mdb) or Microsoft Access project (.adp), click **Bookmark**. Click the [bookmark](#) or object you want in the list, and then click **OK**.
9. Click **OK** in the **Edit Hyperlink** dialog box.

Access adds the hyperlink to the field or control. You can click the hyperlink to open the destination document, Web page, or object.

Notes

- In a [data access page](#), the text box in which you enter a hyperlink address displays the hyperlink address as a text string, not as a formatted hyperlink. The formatted address can be found in the **Hyperlink** property of the text box.
- The [ScreenTip](#) displayed for a hyperlink on a data access page is defined in the hyperlink control, so a ScreenTip that you enter won't be displayed on the page.
- To see a newly created hyperlink address displayed as a formatted hyperlink on the data access page, click **Refresh** on the Microsoft Internet Explorer toolbar.

Troubleshoot hyperlinks

Problems with hyperlinks can be caused by one or more of the following occurrences:

- **The destination may have been moved or may not exist any more.** Verify that the destination exists by viewing the destination file with your Web browser if it is on the Internet, or with Windows Explorer if it is on your hard disk or a network.
- **The text you believe is a hyperlink only looks like a hyperlink.** Select the text and click **Hyperlink**  to make sure the text is a hyperlink.
- **You may not have access to the destination.** If the destination file is on the Internet, make sure you have a connection to the Internet. If the destination is on a network, contact your network administrator to ensure you have access to the destination file.
- **The destination may have been entered incorrectly.** Make sure that the hyperlink destination is correct.

[Show All](#)

Check spelling

[Check the spelling of data in Datasheet view](#)

1. Open a [datasheet](#).
2. Select the records, columns, fields, or text whose spelling you want to check.

[How?](#)

The following table lists mouse techniques for selecting data or records in [Datasheet view](#).

To select	Click
Data in a field	Where you want to start selecting and drag across the data.
An entire field	The left edge of the field in a datasheet, where the pointer changes into  .
Adjacent fields	The left edge of a field and drag to extend the selection.
A column	The field selector .
Adjacent columns	The field name at the top of the column and then drag to extend the selection.
A record	The record selector .
Multiple records	The record selector of the first record, and then drag to extend the selection.
All records	Select All Records on the Edit menu.

3. Click **Spelling**  on the [toolbar](#).

Note If a datasheet contains [subdatasheets](#), Microsoft Access checks the spelling only of the datasheet or subdatasheet where you have placed the insertion point.

[Check the spelling of data in Form view](#)

1. Open a form in [Form view](#).
2. Select the field or text whose spelling you want to check.
3. Click **Spelling**  on the toolbar.

[Check the spelling of the data in a database object while in the Database window](#)

1. In the [Database window](#), click the table, query, form, view, or stored procedure whose spelling you want to check.
2. Click **Spelling**  on the toolbar.

Note In the **Spelling** dialog box, click **Add** if you want to add the word that is in the **Not In Dictionary** box to the custom dictionary.

[Show All](#)

Choose a custom dictionary for checking spelling

1. On the **Tools** menu, click **Options**, and then click the **Spelling** tab.
2. Under **Add words to**, click the [custom dictionary](#) that you want to use.

Note You can't create a custom dictionary in Microsoft Access, but you can create a custom dictionary in Microsoft Word.

Set spelling options

1. On the **Tools** menu, click **Options**, and then click the **Spelling** tab.
2. Select the options you want.

Set the language for checking spelling

1. On the **Tools** menu, click **Options**, and then click the **Spelling** tab.
2. Under **Dictionary Language**, select the language that you want to use when checking spelling.

[Show All](#)

Troubleshoot spelling

[The spelling checker didn't find specific errors.](#)

- Make sure you haven't selected part of the data. If you've selected text, the spelling checker checks only the selected text. When the spelling checker asks whether you want to check the remainder of the document, click **Yes**.
- If the text is in all uppercase letters, make sure that the **Ignore words in UPPERCASE** check box is cleared on the **Spelling** tab (**Tools** menu, **Options** command).

[The spelling checker questions text that is correct.](#)

- Make sure you are using the appropriate dictionary language. If necessary, install the spelling tools for that language, and enable the language for editing.

[How?](#)

1. On the Microsoft Windows **Start** menu, point to **Programs**, point to **Microsoft Office**, point to **Microsoft Office Tools**, and then click **Microsoft Office 2003 Language Settings**.
2. Click the **Enabled Languages** tab.
3. In the **Available languages** box, select the language you want to enable.
4. Click **Add**.

If any Microsoft Office programs are running, the Language Settings tool notifies you and closes them automatically. The features for the language you enabled appear the next time you start an Office program.

Note Some languages— for example, Asian, right-to-left (such as Arabic), and Central European languages— have system requirements that must be met before you can type characters for those languages in Microsoft Office programs. If you see **(limited support)** next to a language in the **Available**

languages box, you need to install system support for that language.

- Add specialized terms or custom capitalizations to a custom dictionary. When you use the spelling checker, it compares the words in your document with those in its main dictionary. The main dictionary contains most common words, but it might not include proper names, technical terms, acronyms, and so on. In addition, some words might be capitalized differently in the main dictionary than in your document. Adding such words or capitalization to a custom dictionary prevents the spelling checker from questioning them.

[I can't select language-specific spelling options.](#)

Make sure you are using the appropriate dictionary language. If necessary, install the spelling tools for that language, and enable the language for editing.

[How?](#)

1. On the Microsoft Windows **Start** menu, point to **Programs**, point to **Microsoft Office**, point to **Microsoft Office Tools**, and then click **Microsoft Office 2003 Language Settings**.
2. Click the **Enabled Languages** tab.
3. In the **Available languages** box, select the language you want to enable.
4. Click **Add**.

If any Microsoft Office programs are running, the Language Settings tool notifies you and closes them automatically. The features for the language you enabled appear the next time you start an Office program.

Note Some languages— for example, Asian, right-to-left (such as Arabic), and Central European languages— have system requirements that must be met before you can type characters for those languages in Microsoft Office programs. If you see **(limited support)** next to a language in the **Available languages** box, you need to install system support for that language.

[I can't add a word to the custom dictionary.](#)

Make sure that the word contains 64 characters or fewer, and that it doesn't contain spaces.

In addition, make sure that the number of words in the custom dictionary does not exceed 5,000, and that its file size does not exceed 64 kilobytes (KB).

[I added words to a custom dictionary, but the spelling checker still questions them.](#)

Before you can use a custom dictionary to check spelling, you must activate it by selecting the check box next to its name in the **Custom Dictionaries** dialog box.

In addition, if you add a word to the custom dictionary while another program is running, the addition does not appear until you restart Microsoft Access.

[Microsoft Access prompts me to create a new custom dictionary when I try to add a word.](#)

By default, Microsoft Access uses an [ANSI](#) code page when it creates a custom dictionary. If you try to add a word while checking spelling, and the language that you are using requires [Unicode](#) support in the custom dictionary, Access keeps the original dictionary intact and prompts you to create a new custom dictionary that can store entries in Unicode.

Set defaults for the appearance of datasheets

1. On the **Tools** menu, click **Options**.
2. Click the **Datasheet** tab.
3. Click the options you want.

Note These options only affect new datasheets. If you want existing datasheets to have the same format also, you must make the changes in the specific datasheets using the **Format** menu commands or remove the saved properties using Microsoft Visual Basic code.

[Show All](#)

Change font, or font style, size, and color in a datasheet

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#).
2. On the **Format** menu, click **Font**.
3. Click the options you want.

Note Changing the format of a specific datasheet overrides the default datasheet format defined in the **Datasheet** tab of the **Options** dialog box (**Tools** menu).

[Show All](#)

Change the gridline style and background color in a datasheet

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#).
2. On the **Format** menu, click **Datasheet**.
3. Click the options you want.

Note Changing the format of a specific datasheet overrides the default datasheet format defined in the **Datasheet** tab of the **Options** dialog box (**Tools** menu). Changing the default datasheet format affects new datasheets, but does not affect existing datasheets.

[Show All](#)

Freeze and unfreeze columns in a datasheet

You can freeze one or more of the columns (fields) on a [datasheet](#) so that they become the leftmost columns and are visible at all times no matter where you scroll.

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#).
2. Select the columns you want to freeze.

To select one column, click the [field selector](#) for that column.

To select more than one column, click the column field selector and then, without releasing the mouse button, drag to extend the selection.

3. Do one of the following:

To freeze the selected columns, click **Freeze Columns** on the **Format** menu.

To unfreeze all columns, click **Unfreeze All Columns** on the **Format** menu.

[Show All](#)

Show or hide columns in a datasheet

1. Open a table, query, form, view, or stored procedure in [Datasheet view](#).
2. Do one of the following:

Hide a column or columns

1. Select the columns you want to hide.

To select one column, click the [field selector](#) for that column.

To select adjacent columns, click a column field selector and then, without releasing the mouse button, drag to extend the selection.

2. On the **Format** menu, click **Hide Columns**.

Show a hidden column or columns

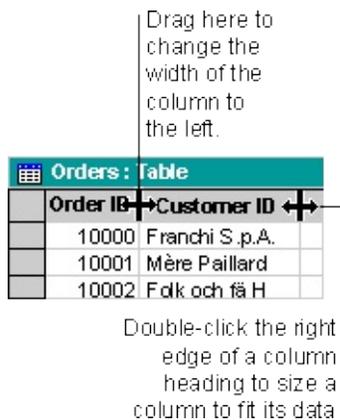
1. On the **Format** menu, click **Unhide Columns**.
2. In the **Unhide Columns** dialog box, select the names of the columns that you want to show.

[Show All](#)

Resize a column or row

1. Open a table, query, or form in [Datasheet view](#).
2. Do one or both of the following:

[Resize a column](#)



Do one of the following:

To resize a column to a specific width, position the pointer on the right edge of the column you want to resize, as shown in the illustration, and drag until the column is the desired size.

To size a column to fit its data, double-click the right edge of the column heading.

[Resize rows](#)



Position the pointer between any two [record selectors](#) at the left side of the datasheet, as shown in the preceding illustration, and drag until the rows are

the desired size.

Note You can't undo changes to the width of columns using the **Undo** command on the **Edit** menu. To undo changes, close the datasheet, and then click **No to All** when asked if you want to save changes to the layout of the datasheet. This will also undo any other layout changes you have made.

[Show All](#)

Troubleshoot datasheets and subdatasheets

Some of the content in this topic may not be applicable to some languages.

Troubleshoot datasheets

 [Access doesn't save formatting changes I make to datasheet in an Access project.](#)

In a [Microsoft Access project](#) that is connected to a database in Microsoft SQL Server 7.0 or earlier or in MSDE 1.0, you can't save datasheet formatting changes unless the datasheet is a form. Datasheet formatting changes include:

- Resizing a column or row
- Showing or hiding columns
- Changing the gridline style
- Changing the background color
- Changing the font or the font style, size, or color

If you need to save formatting changes to an Access project table, view, or stored procedure datasheet, use the AutoForm Wizard to create the datasheet as a form (which quickly sets the **RecordSource** property to the datasheet and adds all the fields to the form), and then set the form's **DefaultView** and **ViewsAllowed** properties to **Datasheet** (which makes the form behave like a datasheet).

Troubleshoot subdatasheets

[I can't insert a subdatasheet in a datasheet in an Access project.](#)

You can't use subdatasheets in a Microsoft Access project that is connected to a database in Microsoft SQL Server 7.0 or earlier or in MSDE 1.0.

[I can't see the foreign key or matching fields in my subdatasheet.](#)

By default, Microsoft Access does not display the [foreign key](#) or matching field in the [subdatasheet](#) of a table or query. You can temporarily show the foreign key or matching field by using the **Unhide Columns** command (**Format** menu), but the next time you open the datasheet containing the subdatasheet, the column will be hidden again. Hiding or showing this field changes only the appearance of the subdatasheet; it does not modify any [relationship](#).

In a form, you can permanently display the foreign key or matching field in a subdatasheet corresponding to a subform if the subform's source object is a form.

[My datasheet containing a subdatasheet looks different from when I last saved it.](#)

The **SubdatasheetExpanded** property controls the state of a [subdatasheet](#) in a table or query, or the [subform control](#) in a form. You can set the property to **Yes** (all rows expanded) by clicking **Expand All** (**Format** menu, **Subdatasheet** submenu) or **No** (all rows collapsed) by clicking **Collapse All** (**Format** menu, **Subdatasheet** submenu). Clicking one or more [expand indicators](#) to show subdatasheets for one or more rows in the datasheet doesn't change the **SubdatasheetExpanded** property; even if you click all expand indicators to show all subdatasheets, the property is still set to **No**, and all rows are collapsed the next time you open the datasheet.

[My subdatasheet displays the same data for each row of the datasheet.](#)

In a table or query, you need to specify the matching field between the datasheet and [subdatasheet](#) in the **List Child Fields** and **Link Master Fields** boxes in the **Insert Subdatasheet** dialog box (**Insert** menu, **Subdatasheet** command). These

boxes correspond to the **LinkChildFields** and **LinkMasterFields** properties of the datasheet containing the subdatasheet.

In a form, you need to specify the matching field between the [main form](#) and the subform in the **LinkChildFields** and **LinkMasterFields** properties of the [subform control](#) on the main form.

[The **Link Master Fields** box is empty when I add a subdatasheet to a query in a Microsoft Access database.](#)

If you have just created the query, you must save it first and then add the [subdatasheet](#) to see the fields from the query in the **Link Master Fields** box.

[I have defined a relationship between two tables in a Microsoft Access database but I don't see a subdatasheet.](#)

When the **SubdatasheetName** property of a table is set to **Auto** and the table is on the "one" side of a [one-to-many relationship](#) or is in a [one-to-one relationship](#), Microsoft Access automatically creates a [subdatasheet](#). However, the value of the **SubdatasheetName** property can be set to a specific table or query in the database; or it can be set to **None**, in which case no subdatasheet will be created. To return to the default behavior, set the value of the **SubdatasheetName** property to **Auto**.

[Show All](#)

About subdatasheets

In a [subdatasheet](#), you can view and edit [related](#) or [joined](#) data in a table, query, or form datasheet, or in a [subform](#). For example, in the Northwind sample database, the Suppliers table has a [one-to-many relationship](#) with the Products table; so for each row of the Suppliers table in [Datasheet view](#), you can view and edit the related rows of the Products table in a subdatasheet.



- 1 Click the expand indicator to expand or collapse a subdatasheet

Microsoft Access automatically creates a subdatasheet in a table that is in a [one-to-one relationship](#), or is on the "one" side of a one-to-many relationship, when the **SubdatasheetName** property of the table is set to **Auto**. A relationship is defined by matching [primary key](#) and [foreign key](#) fields in the related tables.

However, you can add a subdatasheet to any table, query, or form. A subdatasheet can have as its source object a table or query (specified in the **SubdatasheetName** property). A subdatasheet corresponding to a [subform control](#) can have as its source object a table, query, or form (specified in the subform control's **SourceObject** property).

You can nest a subdatasheet within a subdatasheet for up to eight levels. However, each datasheet or subdatasheet can have only one nested subdatasheet. For example, the Customers table can contain one Orders table subdatasheet, and the Orders table subdatasheet can contain one Orders Details subdatasheet. But the Customers table can't contain both an Orders table subdatasheet and a Salespeople query subdatasheet.

[Forms, subforms, and subdatasheets](#)

A form in [Form view](#) or Datasheet view can also have a subdatasheet, but there are differences between a form subdatasheet and a table or query subdatasheet:

- You insert a subdatasheet into a form by adding a subform control, and then defining a source object by setting the subform control's **SourceObject** property to a table, query, or form. When you view the form in Datasheet view, the subform appears as a subdatasheet. If the subform's source object is a form, its **DefaultView** property can be set to **Datasheet**, **Single View**, or **Continuous Form**. Therefore, in form Datasheet view, you can also see the subdatasheet displayed as a single or continuous form.
- Although a form can have more than one subform, each of which can be nested for up to two levels, in form Datasheet view only one subdatasheet is displayed. Access displays the subdatasheet corresponding to the first subform in the tab order of the form.
- When you first open a table, query, or form in Datasheet view, its subdatasheet is not expanded; however, when you first open a form in Form view, the subdatasheet corresponding to the subform is expanded.
- You can permanently display the foreign key or matching fields in the subdatasheet corresponding to the subform if the subform's source object is a form.

[Show All](#)

Insert or modify a subdatasheet

1. Open a table or query in [Datasheet view](#).
2. On the **Insert** menu, click **Subdatasheet**.
3. Click the **Tables**, **Queries**, or **Both** tab, depending on which kind of object you want to insert as a subdatasheet.
4. Click a table or query in the list.
5. In the **Link Child Fields** box, click the field to be the [foreign key](#) or matching field for the table or query that is the subdatasheet you chose in step 4.
6. In the **Link Master Fields** box, click the field to be the [primary key](#) or matching field for the table or query that is the main datasheet you opened in step 1.

Note If you want to specify two or more fields in the **Link Child Fields** or **Link Master Fields** box, enter the field names separated by a semicolon (;) — for example, **Last Name;First Name**.



You can create a self-join when you insert or modify a subdatasheet. For example, suppose you want to see the employees who report to each manager as a subdatasheet of the Employees table. Open the Employees table, click **Subdatasheet** on the **Insert** menu, click the **Tables** tab, and click the Employees table. Then click ReportsTo in the **Link Child Fields** box, and click EmployeeID in the **Link Master Fields** box.

[Show All](#)

Remove a subdatasheet

1. Open the table or query containing the subdatasheet in [Datasheet view](#).
2. On the **Format** menu, point to **Subdatasheet**, and then click **Remove**.

Note Removing a subdatasheet does not delete the table or query, nor does it modify any [relationship](#).

[Show All](#)

Expand or collapse a subdatasheet

1. Open the table, query, or form containing the [subdatasheet](#) in [Datasheet view](#).
2. Do one of the following:

To expand a subdatasheet for a row of a datasheet or subdatasheet, click the [expand indicator](#) (+) to the left of the row.

To expand all subdatasheets in the datasheet or subdatasheet that currently has focus, point to **Subdatasheet** on the **Format** menu, and then click **Expand All**.

To collapse a subdatasheet for a row of a datasheet or subdatasheet, click the expand indicator (-) to the left of the row.

To collapse all subdatasheets in the datasheet or subdatasheet that currently has focus, point to **Subdatasheet** on the **Format** menu, and then click **Collapse All**.

[Show All](#)

Show or hide the foreign key in a subdatasheet

By default, Microsoft Access doesn't display the [foreign key](#) or the matching field in a [subdatasheet](#). However, you can show the foreign key field temporarily.

1. In the datasheet, click the [expand indicator](#) (+) and then select a record in the subdatasheet.
2. On the **Format** menu, click **Unhide Columns**.
3. Select the check box next to the foreign key field.

Notes

- The next time you open the datasheet containing the subdatasheet, the column or columns will be hidden again.
- Hiding or showing these fields changes only the appearance of the subdatasheet; it does not modify any [relationship](#) or source object.

[Show All](#)

About designing a PivotTable or PivotChart view

When you first open a datasheet or form in [PivotTable](#) or [PivotChart view](#), the view does not include fields from the underlying [record source](#) or form. The following illustration shows the Employees table when opened for the first time in PivotTable or PivotChart view.



[Adding a PivotTable view](#)

Use the field list to view the list of fieldsets and fields that are available from the source data and to add fields to the [row](#), [column](#), [filter](#), and [detail areas](#) of the PivotTable view. In the field list, the fields that are not bold are available but not yet displayed in the PivotTable view. The bold fields are already displayed. You can scroll and click the expand indicators (⊕ or ⊖ boxes) next to the fieldsets to show or hide fields in the field list.

The following illustration shows the Employees table in PivotTable view. The view shows details of employees in each city.

City	Seattle			GrandTotal
HireDate	5/1/1992	3/5/1994		
Title	LastN	LastN	No Totals	
Inside Sales				
Sales Repre.	Davolio	Callahan		
Grand Total				

- 1 City field is in the filter area.
- 2 Title field is in the row area.
- 3 Hire Date field is in the column area.
- 4 Last Name field is in the detail area.

[Adding fields to the detail area](#)

Use [detail fields](#) to display detail data and see what data is available to be summarized. When you add fields to the detail area, you display all of the detail records from the source data for the fields. Field names become column labels, and the detail records are displayed in rows below the column labels.

Sport	Quarter
Golf	Qtr3
Golf	Qtr4
Tennis	Qtr3

- 1 Detail fields
- 2 Detail data

You can also add a calculated detail field that uses an [expression](#) to calculate detail values.

[Adding fields to the row or column area](#)

Use [row](#) and [column fields](#) to summarize and compare data. These fields display the unique items of data within a field down rows or across columns. The cell at each row and column intersection summarizes the data for an item.

	Quarter ▼	
	Qtr3	
	+ -	
Sport ▼	Sales	
Golf		\$7,930
Tennis		\$4,670

- 1 Column field
- 2 Column field item
- 3 Row field
- 4 Row field items

[Adding fields to the filter area](#)

Use a [filter field](#) to confine the view to a particular part of the available data. When you select an item in a filter field, data is displayed and calculated only for that item. For example, if you add a Region filter field, you can have the PivotTable view display and calculate data for the regions you select.

Region ▼		
East		
Sport ▼	Sales	
Golf		\$7,930
Tennis		\$4,670
Grand Total		\$12,600

- 1 Filter field
- 2 Filter field item
- 3 Summary for the selected region

[Adding total fields](#)

You can create new fields by adding [total fields](#) to the PivotTable view. Total fields summarize detail data. For example, if the datasheet or form includes a Sales field that shows the amount of each sale, you could add a Sum of Sales total field to calculate the total sales, or a Count of Sales total field to calculate

how many units were sold.

You can also add a calculated total field that uses an [expression](#) to calculate total values.

[Removing fields](#)

To eliminate data that you don't want to see or summarize in the view, you can remove fields from the PivotTable view layout. Removing a field does not remove it from the available source data. A field that you remove is no longer displayed in the PivotTable view, but it remains available in the field list for viewing.

[Moving fields](#)

After you add a field to a PivotTable view, you can move it to other areas. When you move a field, you change the layout of the PivotTable view, and you change the data that's displayed and the summaries that are calculated.

[Impact of adding and removing fields on filtering](#)

Filter settings are retained when you remove a field. If you remove a field and later add the field back to the layout, the same items are again hidden.

[Impact of your changes on PivotChart view](#)

When you add, move, or remove fields in PivotTable view, Microsoft Access automatically updates the PivotChart view to reflect the change you made. For example, if you add a [filter field](#) to the PivotTable view, Access adds the field to the filter area in PivotChart view as well. Similarly, if you make a change in PivotChart view, Access will update the PivotTable view automatically. So, after you design the layout of one view and switch to the other view for the first time, you will not see a blank view.

Access does not synchronize formats between PivotTable and PivotChart views.

[Saving your changes](#)

If you have made changes to the layout of a form in PivotTable or PivotChart view, Microsoft Access automatically saves your changes when you close the object. If you have made changes to the layout of a datasheet in one of these views, Access asks whether you want to save your changes before quitting.

Layout information is not stored separately for each user. For example, if another user opens a form in PivotChart view and makes changes to the layout, the next time you open the form in PivotChart or PivotTable view, you will see the layout as it was modified by the other user.

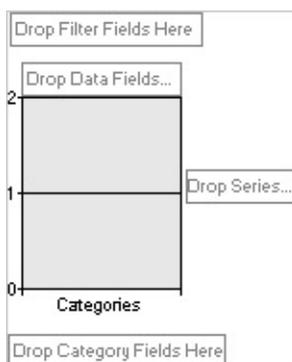
[Adding a PivotChart view](#)

Use the field list to view the list of fieldsets and fields that are available from the source data and to add fields to the category, series, filter, and MultiChart areas of the PivotChart view. In the field list, the fields that are not bold are available but not yet displayed in the PivotChart view. The bold fields are already displayed. You can scroll and click the expand indicators (⊕ or ⊖ boxes) next to the fieldsets to show or hide fields in the field list.

[Using the drop areas](#)

You change the layout of a PivotChart view by moving the fields to predefined [drop areas](#) within the chart workspace. To do this, drop areas must be shown in the chart.

Drop areas are displayed differently depending on the chart type. For example, a series drop area is not displayed for a pie chart because pie charts consist of only one series. If you plot multiple charts, you will see a drop area for multi-chart fields. This type of drop area is not displayed for single charts.



After you move fields to the drop areas so that the drop-area captions are covered up, you can still drag additional fields to the areas.

[Moving fields to the category and series areas](#)

When you move a field to the series area, the unique items of data within the field are displayed as data series in the chart. These series are represented by colored [data markers](#), and their names appear in the chart [legend](#).

When you move a field to the category area, the unique items of data are displayed as categories, or related groups of data. Each category consists of one point from each data series. Category labels usually appear across the x axis of the chart, although this can vary depending on the type of chart you are using.

[Moving fields to the data area](#)

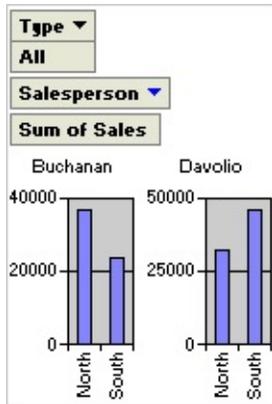
Data fields provide the values to be summarized in the chart. When you move a field to the data area, the values from the field are used as the data that is measured in the chart.

[Moving fields to the filter area](#)

A filter field is similar to a page field in a Microsoft Excel PivotTable report. Filter fields allow you to confine the view to a particular part of the available data. For example, when you move a Product field to the filter area, you can have the chart display category and series values for one product at a time.

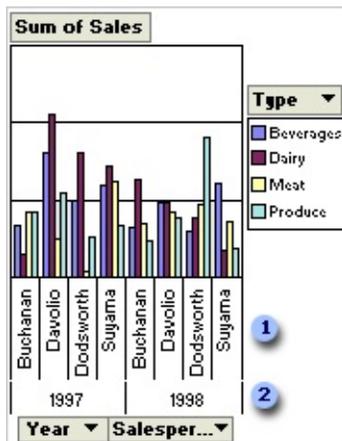
[Moving fields to the MultiChart area](#)

When your chart is based on data from a PivotTable list or from a database table or query, you can plot multiple charts. When you move a field to the MultiChart area, items in that field become separate charts. For example, if you move the Salesperson field to the MultiChart area, a chart is created based on data for each salesperson in that field. In the following example, the Salesperson field is in the MultiChart area, but it's filtered so it displays individual charts for Buchanan and Davolio.



[Moving the category or series fields to inner or outer levels](#)

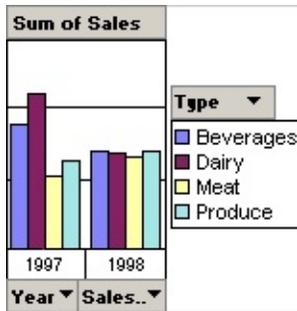
When a chart contains multiple [series](#) or [category fields](#), the fields that are closest to the data are referred to as inner fields. The other fields are outer fields. In the following example, Year is the outer field and Salesperson is the inner field. The inner field items are displayed as salespeople's names, and the outer field items are displayed as the years 1997 and 1998.



1 Inner field items

2 Outer field items

You can expand and collapse multiple fields to show more or less information in a particular field. For example, you can collapse the outer field (Year) in the example so that the inner field items are no longer displayed.



[Adding and removing fields](#)

The layout of a chart does not have to include all fields that are available from the source data. If more data is available, you can add fields to the chart. For example, if the chart summarizes sales revenue, and the source data also includes sales quantities, you might add the Quantity field as a data field to summarize both revenue and quantity of products sold. You can also remove fields from the chart layout that you no longer want to see.

[Impact of changing layout on filtering](#)

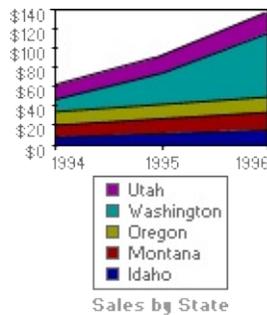
Filter settings are retained when you move or remove a field. This means that when you move a [series](#) or [category field](#) to the filter area and back, previously hidden items are again hidden. If you remove a field and later add the field back to the layout, the same items are again hidden.

[Show All](#)

Examples of chart types in PivotChart view

Area

An area chart emphasizes the magnitude of change over time. By displaying the sum of the plotted values, an area chart also shows the relationship of parts to a whole.



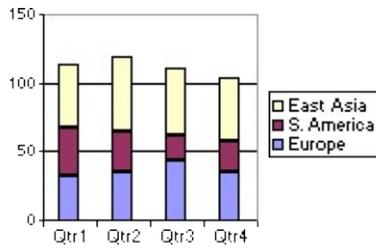
In this example, an area chart emphasizes increased sales in Washington and illustrates the contribution of each state to total sales.

Column

A column chart shows data changes over a period of time or illustrates comparisons among items. To emphasize variation over time, categories are organized horizontally and values are organized vertically.

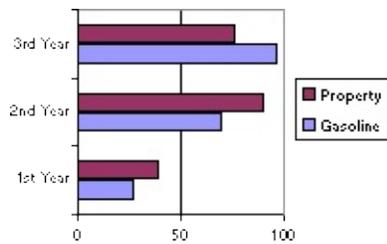


Stacked column charts show the relationship of individual items to the whole.

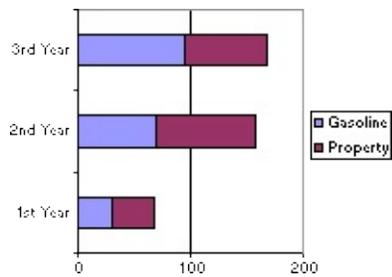


Bar

A bar chart illustrates comparisons between individual items. To focus on comparing values and to place less emphasis on time, categories are organized vertically and values are organized horizontally.

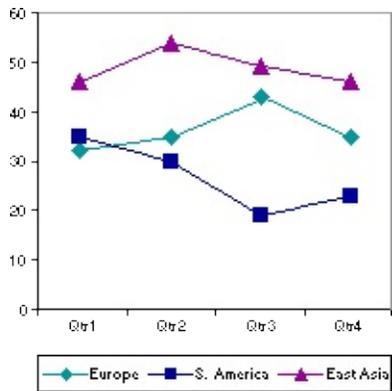


Stacked bar charts show the relationship of individual items to the whole.



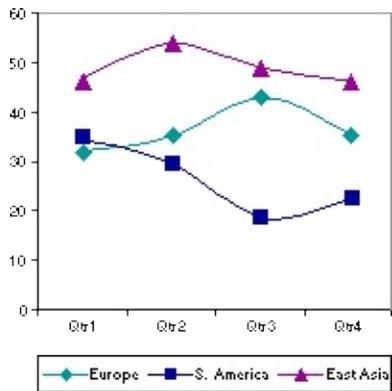
Line

A line chart shows trends in data at equal intervals.



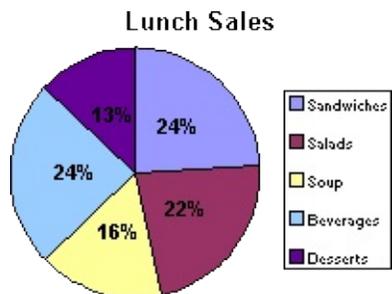
SmoothLine

A smooth line chart shows trends in data at equal intervals, smoothed to show the estimated data between intervals.



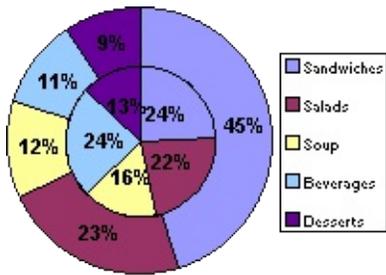
Pie

A pie chart shows the proportional size of items that make up a [data series](#) to the sum of the items. It always shows only one data series and is useful when you want to emphasize a significant element.



You can also create a stacked pie chart, which is a pie chart with more than one

series.

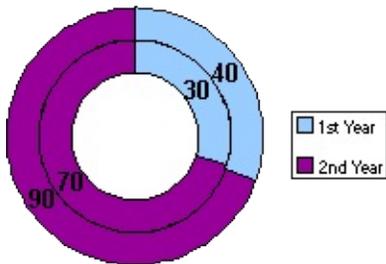


Units Sold (inner pie) and Gross Sales (outer pie)

Doughnut

Like a pie chart, a doughnut chart shows the relationship of parts to a whole, but a doughnut chart consists of rings and can contain more than one [data series](#). Each ring of the doughnut chart represents a data series.

Revenue from Gasoline and Property Taxes

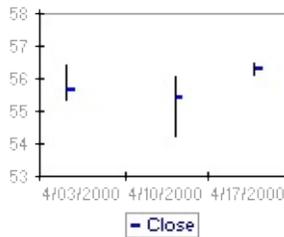


Property (outer ring) Gasoline (inner ring)

Stock

Two stock chart types are often used to illustrate stock prices: open-high-low-close and high-low-close. These charts can also be used for scientific data—for example, to indicate high-low-average temperatures. The chart in the example is a high-low-close chart.

Date	High	Low	Close
4/03/2000	56 3/8	55 1/4	55 5/8
4/10/2000	56 1/8	55 1/2	55 1/2
4/17/2000	56 3/8	56	56 1/4

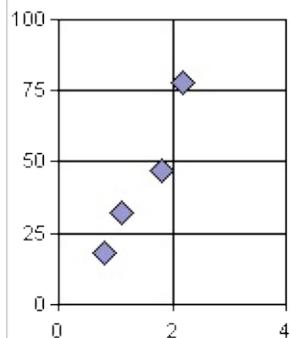


XY (Scatter)

An xy (scatter) chart either shows the relationships among the numeric values in several [data series](#) or plots two groups of numbers as one series of xy coordinates. This chart shows uneven intervals— or clusters— of data and is commonly used for scientific data.

When you arrange your data, place the x values in one row or column with corresponding y values in the adjacent row or column in an x-y x-y arrangement.

X Value 1	Y Value 1
Volume	Temperature
0.8	18
1.1	32
1.8	47
2.2	78



Bubble

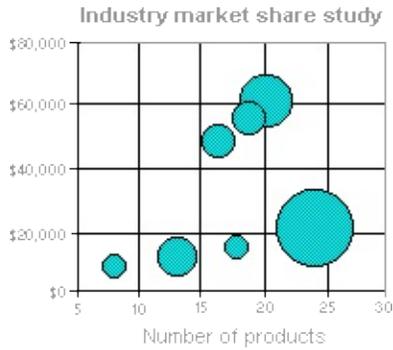
A bubble chart is a type of xy (scatter) chart. The size of the [data marker](#) indicates the value of a third variable.

To arrange your data, place the x values in one row or column and then enter the

corresponding y values and bubble sizes in the adjacent rows or columns.

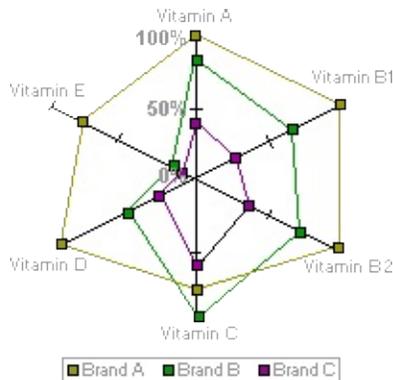
No. of products	Sales	Market share %
14	\$11,200	13
20	\$60,000	23
18	\$14,400	5

X values Y values Bubble sizes



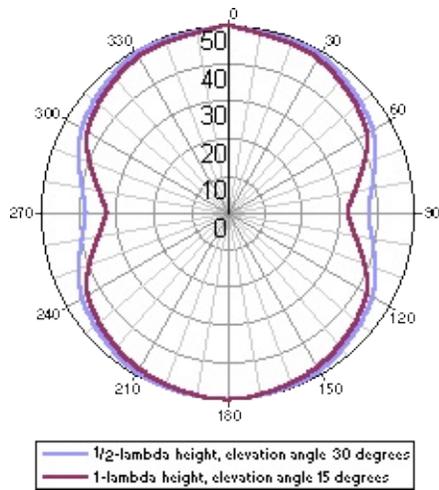
Radar

In a radar chart, each category has its own value axis that radiates from the center point. Lines connect all the values in the same [data series](#).



A radar chart compares the aggregate values of several data series. In this chart, the data series that covers the most area, Brand A, represents the brand with the highest vitamin content.

Polar



A polar chart compares the relationship of angles and distances. The chart in this example illustrates elevation angles for two heights of a radio antenna.

[Show All](#)

Elements of a PivotTable or PivotChart view

[Elements of a PivotTable view](#)

Reports To		Country		Region		Grand Total	
All		UK	USA	W.A.	Total	Total	Total
Title		Last Name		No. Emp		No. Emp	
Sales Mgr	Buchanan						1
Total		1				1	
VP Sales		Fuller				1	
Total		1				1	
Grand Total		1		1		2	

- 1 Filter field
- 2 Row fields
- 3 Column fields
- 4 Field drop-down arrow
- 5 Detail field
- 6 Expand indicators

[Fields](#)

Each field corresponds to a column of data and can summarize multiple rows of information. Types of fields include row fields, column fields, detail fields, and filter fields.

[Item](#)

A unique element of data within a field. Items of data in [row](#) and [column fields](#)

are listed down rows and across columns. The cell where a row and column intersect displays summarized data for the items listed at the top of the column and the left side of the row. When a lower level of items is available for display, an expand indicator  appears beside the item. The items at the lowest level don't display the expand indicator. The expand indicators ( and  boxes) to the right of an item indicate that detail data is available for that item.

[Row field](#)

A field in the [row area](#). In the above PivotTable view, Title and Hire Date are row fields. Items in row fields are listed down the left side of the view. Title is the outer row field, and Hire Date is the inner row field.

[Column field](#)

A field in the [column area](#). Items in column fields are listed across the top of the view. In the above PivotTable view, Country and Region are column fields, where Country is the outer column field and Region is the inner column field.

[Filter field](#)

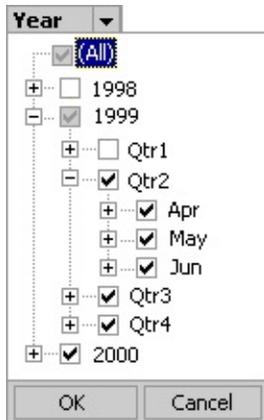
A field in the [filter area](#) that you can use to display data for specific items. In the above PivotTable view, Reports To is a filter field that you can use to display data for all managers or only selected managers.

[Detail field](#)

A field in the [detail area](#) that displays all of the rows, or records, from the underlying [record source](#). You can display detail rows and summarized data simultaneously.

[Field drop-down arrow](#)

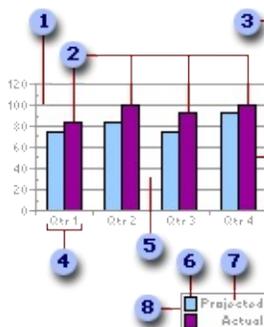
The arrow  at the right side of each field. Click this arrow to display a list from which you can select the items to display and hide. Date and time fields and fields with [custom groups](#) provide different levels in all types of source data. For example, clicking the arrow next to a Year field might display the following:



The cleared check box next to **1998** indicates that the year 1998 is hidden. The shaded check box next to **1999** indicates that the year 1999 is currently selected for display but some of the items under **1999** are not selected for display. Within **1999**, **Qtr1** is hidden. The selected check box next to **2000** indicates that the year 2000 is selected for display. Selecting **(All)** selects all items at all levels for display.

[Elements of a PivotChart view](#)

[Series, categories, and data representation](#)



- 1 Gridline
- 2 Data series
- 3 Data marker
- 4 Category
- 5 Plot area

6 Legend key

7 Series name

8 Legend

Gridlines Lines you can add to a chart that make it easier to view and evaluate data. Gridlines extend from the tick marks on an axis and then continue across the plot area.

Data series, series name (legend entry) A group of related data points plotted in a chart. Each data series has a unique color or pattern and is represented in the legend by the legend key and series name (legend entry). In the preceding chart example, Projected and Actual are data series. You can plot one or more data series in most charts.

Data marker Used to represent a data point, which is an individual value plotted in a chart. Related data points constitute a data series. Examples of data markers include bars, dots, and slices.

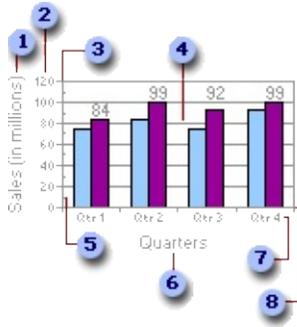
Category A group of related data points made up of one data point from each data series in the chart. For most charts, categories are plotted along the category (x) axis, which is usually horizontal. However, in some charts the category (x) axis is vertical, as in a bar chart. In the preceding chart example, Qtr1, Qtr2, Qtr3, and Qtr4 are categories, each made up of one point from the Projected series and one point from the Actual series.

Plot area The area in a chart that's bounded by the axes and includes all the data series.

Legend key Images in a legend that show the patterns and colors assigned to the data series or categories in a chart. Legend keys appear to the left of legend entries.

Legend A key to the patterns or colors assigned to the data series or categories in a chart.

 [Axes](#)



- 1 Value (y) axis title
- 2 Value (y) axis tick mark labels
- 3 Value (y) axis
- 4 Gridline
- 5 Minor tick mark
- 6 Category (x) axis title
- 7 Category (x) axis tick mark labels
- 8 Category (x) axis

Value (y) axis title A title that is usually used to describe what the value axis measures.

Value (y) axis tick mark labels Also known as value axis labels. These labels describe the values of measurement on the value axis.

Value (y) axis A line that borders one side of the plot area, providing a frame of reference for measurement or comparison in a chart. For most charts, data values are plotted along the value (y) axis, which is usually vertical. However, in some charts the value (y) axis is horizontal, as in a bar chart. Some charts have data values plotted on both axes, as in an xy (scatter) chart.

Gridline Lines you can add to a chart that make it easier to view and evaluate data. Gridlines extend from the tick marks on an axis and then continue across the plot area.

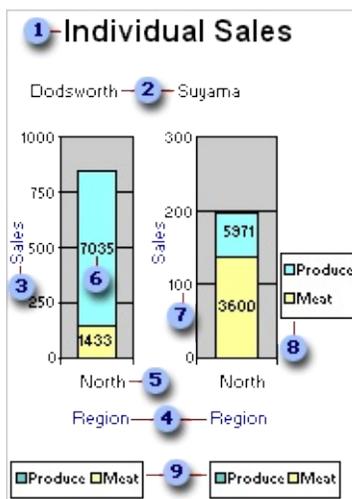
Major and minor tick marks Small lines of measurement that intersect an axis, similar to divisions on a ruler. Major tick marks appear at each major unit of measure on the axis scale, and minor tick marks appear at each minor unit of measure. Both types can be positioned inside or outside of the axis, or they can be positioned to cross the axis.

Category (x) axis title A title that is usually used to describe what is plotted on the category axis.

Category (x) axis tick mark labels Also known as category labels. These labels show the names of the categories on the category axis.

Category (x) axis A line that borders one side of the plot area, providing a frame of reference for measurement or comparison in a chart. Categories are groups of data points made up of one data point from each data series in the chart. For most charts, categories are plotted along the category (x) axis, which is usually horizontal. However, in some charts the category (x) axis is vertical, as in a bar chart. Some charts do not have a category (x) axis, as in an xy (scatter) chart.

[Titles and labels](#)



- 1 Chart workspace title (title for all charts in workspace)
- 2 Chart title (title for single charts)
- 3 Value axis title

- 4 Category axis title
- 5 Category label
- 6 Data label
- 7 Value axis label
- 8 Chart workspace legend (legend for all charts in workspace)
- 9 Chart legend (legend for single chart)

Chart workspace title A title that describes all charts when you have multiple charts in the workspace. This title can be positioned above, below, to the left, or to the right of the chart workspace.

Chart title The title that describes a single chart. This title can be positioned above, below, to the left, or to the right of the chart.

Value axis title A title that is usually used to describe what the value axis measures.

Category axis title A title that is usually used to describe what is plotted on the category axis.

Category label Also known as a tick mark label. These labels show the names of categories on the category axis. A category is a group of related data points, made up of one data point from each [data series](#) in the chart.

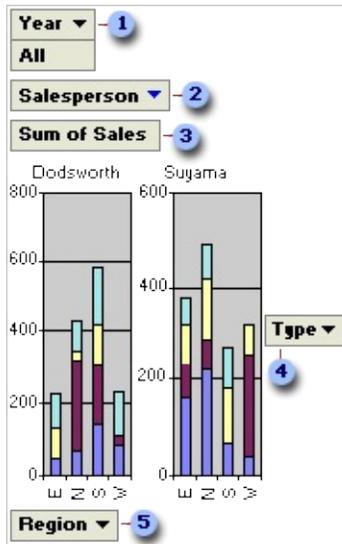
Data label A label that provides additional information about a [data marker](#), which represents a single data point or value. Depending on the chart type, data labels can show values, names of data series or categories, percentages, or a combination of these.

Value axis label Also known as a tick mark label. These labels describe the values of measurement on the value axis.

Chart workspace legend A key to the patterns or colors assigned to the data series or categories in all charts in a multiple-chart workspace.

Chart legend A key to the patterns or colors assigned to the data series or categories in a single chart.

Fields



- 1 Filter field
- 2 MultiChart field
- 3 Data field
- 4 Series field
- 5 Category field

Filter field A field that you use to filter data by specific items. In the previous example, the Year filter field displays data for all years for which there are records. To display data for a single year, you can click the arrow next to **Year** and select the year.

MultiChart field A field you use to display separate charts for each item in the field. In the example, the Salesperson field is filtered to show two salespeople: Dodsworth and Suyama. Because the field is in the MultiChart area, a separate chart is displayed for each.

Data field A field from the underlying source data that provides values to

compare or measure. In the example, Sum of Sales is a data field that summarizes sales in each region for each type of product. The first category [data marker](#) in the chart for Dodsworth (E) reaches a little over 200 on the value (y) axis. This amount is the sum of Dodsworth's sales for products in the East region. Depending on the source data you use for the report, you can change the summary function from **Sum** to **Average**, **Max**, **Min**, or another calculation.

Series field A field that you assign to a series orientation in a chart. The items in the field provide the individual [data series](#). In the example, Type is a series field.

Category field A field from the source data that is assigned to a category orientation in a chart. A category field provides the individual categories for which data points are charted. In the example, Region is a category field.

[Show All](#)

Select an element in PivotTable or PivotChart view

[Select an element in PivotTable view](#)

1. Open a datasheet or form in PivotTable view.
2. Do one of the following:

[Select one or more data cells](#)

- To select a single cell, click it. To select a group of cells, click and drag the mouse pointer until all the cells you want are selected.

[Select a row, column, filter, detail, or total field](#)

- Click the field label.

Notes

- Selecting a [row](#) or [column field](#) also selects all of the items of data in the field.
- If detail data is displayed for several items so that the field label is displayed more than once, clicking one instance of the field label selects all instances.
- Formats that you apply affect the data cells of all total fields in the view.

[Select one or more items](#)

- To select a single [item](#), click the item label. To add more items to the selection, press and hold the CTRL key when clicking the item labels.
- To select a range of items, click the label of the first item, press SHIFT, and

then click the label of the last item.

Notes

- Selecting an item selects all of the data for that item.
- Formats that you apply to one item affect all other items in the field.

[Select a subtotal or grand total](#)

- Click the subtotal or grand total.

Note Selecting a subtotal or grand total also selects all of the data for the subtotal or grand total.

[Select a drop area](#)

1. Click **Properties**  on the **PivotTable** toolbar, and then click the **Captions** tab.
2. In the **Select caption** list, select the item you want.

Note Formats that you apply to a [drop area](#) do not affect other drop areas.

[Select an element in PivotChart view](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click once on the item in the chart you want to select. To select an entire group of items, such as an entire [data series](#), click a second time.

Note You can also select items in the **Select** box on the **General** tab of the **Properties** dialog box (click **Properties**  on the **PivotChart** toolbar). For example, to select the first value axis on the chart, click **Value Axis 1** in the **Select** box.

[Show All](#)

Show or hide the Properties dialog box in PivotTable or PivotChart view

Use the **Properties** dialog box to do common tasks, such as filtering data, formatting text, and showing or hiding elements.

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. Right-click the view, and then click **Properties**  on the shortcut menu.

Note The tabs displayed in the dialog box change as you select different elements.

[Show All](#)

Troubleshoot PivotTable or PivotChart view

[Troubleshoot PivotTable view](#)

[I can't filter data the way I want to.](#)

In [PivotTable view](#), you cannot filter based on part of the data in a cell. This feature is available in [Form view](#) or [Datasheet view](#).

[When I change the layout, data takes a long time to appear.](#)

If the [PivotTable view](#) takes a long time to calculate, you can make several changes to the layout at one time and then retrieve the data when you are finished making changes.

1. Press ESC to cancel the calculation. The view then displays only the names of the fields.
2. Change the layout, adding and moving as many fields as you want.
3. To retrieve and calculate the data for the new layout, click **Refresh**  on the **PivotTable** toolbar.

[I can't display items or detail data.](#)

Expand indicators might be turned off If you don't see the  boxes to the left of items of data in row and column fields, make sure the [expand indicators](#) are turned on.

Items might not be available When an item of data is at the lowest level, the item does not have the  box beside it, even when you turn on the expand indicators.

[Fields are present, but all the data is missing.](#)

You canceled a refresh operation. To display the data, click **Refresh**  on the **PivotTable** toolbar.

 [Tip](#)

Before you refresh the data, you can change the layout or filter the [items](#) in the fields.

 [Data is missing after I copy data in PivotTable view into Excel or Word.](#)

When you copy data in a [PivotTable view](#) to the Clipboard and paste it into Microsoft Excel or Microsoft Word, only visible rows and columns are copied. Any data that you have hidden or filtered out is not copied.

Before you copy data, turn off filtering to show all data in the PivotTable view, and display the detail data that you want copied.

To avoid having to show all the data first, you can export, rather than copy, the contents of the PivotTable view. When you export a copy of the view to Excel, all of the data is automatically included, with the same filtering in effect. You have the same access to additional fields in the underlying record source that you do in the PivotTable view, and you can change the filtering in Excel. From Excel, you can then copy data to Word or other programs. For information about working with PivotTable reports in Excel and publishing them to a Web page, see Help in Excel.

 [A cell displays ##### instead of a number.](#)

When a number is too wide to fit in a cell, the [PivotTable view](#) displays number signs instead of the number. You can fix this in any of the following ways:

- Make the column wider so that the entire number fits.
- Make the font smaller if you don't want to widen the column.
- Change the number format to one that takes up less space.

 [Total fields in my PivotTable view have asterisks next to them.](#)

When you set the option to include all items in totals, asterisks (*) appear next to the caption of the [total fields](#) to indicate that the totals include any hidden items in addition to the displayed items.

If you want to omit data that is hidden by filtering from subtotals and grand totals, you can set an option to include only displayed data. This option also hides the asterisks.

[Troubleshoot PivotChart view](#)

[I can't find the command or option I want.](#)

If you cannot locate the command or option you want in the **Properties** dialog box, it could be because you don't have the necessary item selected in the chart. To make sure you have the correct item selected, click the **General** tab in the **Properties** dialog box, and then in the **Select** box, click the chart item you want to work with. Options for that chart item will then appear in the **Properties** dialog box.

[Text is missing along the category \(x\) axis of the chart.](#)

There might not be enough room in the chart to display all of the axis labels. If some of the category names aren't visible along the category (x) axis of the chart, try the following:

- Use a smaller font size for text on the category (x) axis.
- Shorten the category names in the source data.

[My xy \(scatter\) chart does not use the right values along the category \(x\) axis.](#)

If your category (x) axis displays numbers such as 1, 2, 3, and 4 instead of the values you want, you might have created a line chart instead of an xy (scatter) chart. Click **Chart Type**  on the **PivotChart** toolbar, and then click **XY (Scatter)** in the **Chart type** box. If you want lines to connect the data markers, click one of the subtypes that has lines.

[I can't move or resize chart items with the mouse.](#)

Chart items (such as the legend, plot area, data labels, and titles) can't be moved or resized with the mouse. However, many chart items can be moved by using the **Position** or **Placement** settings on the appropriate tabs in the **Properties** dialog box. For example, you can rotate or flip the plot area of a chart, and you can pull out slices in pie or doughnut charts. You can also resize the font that's used in axis labels, data labels, titles, and the legend; and you can resize the data markers in some types of charts. Other chart items are resized automatically when the size of the chart changes.

[I can't add text to a chart.](#)

You can add titles to a chart and to the axes in a chart. You can also add data labels to [data markers](#). You can't add free-floating text to the chart.

[Show All](#)

About customizing the layout of a PivotTable or PivotChart view

[Customizing the layout of a PivotTable view](#)

You can control how the data is presented in a [PivotTable view](#) by customizing the layout. A different layout lets you calculate and compare summarized values for different elements in your data, or display summaries for a subset of the data.

[Using the drop areas](#)

You change the layout by moving the fields to predefined drop areas within the PivotTable view workspace.

The screenshot shows a PivotTable workspace with three drop areas: 'Drop Filter Fields Here' (top), 'Drop Column Fields Here' (middle), and 'Drop Row Fields Here' (left). A table is displayed with the following data:

	Sport	Quarter	Sales
	Golf	Qtr3	\$1,500
	Golf	Qtr4	\$2,000
	Tennis	Qtr3	\$600
	Tennis	Qtr4	\$1,500
	Tennis	Qtr3	\$4,070
	Tennis	Qtr4	\$5,000
	Golf	Qtr3	\$6,430

1 Drop areas

2 Detail area

After you move fields to the drop areas so that the drop area captions are covered up, you can still drag additional fields to the areas.

[Moving fields to the row or column areas](#)

When you move a field to the [row area](#), the unique items of data within the field are displayed down the rows of the PivotTable view. When you move a field to the [column area](#), the unique items of data are displayed across the columns.

[Moving fields to the detail area](#)

To view total or detail data from the underlying [record source](#), move fields to the [detail area](#).

[Moving fields to the filter area](#)

[Filter fields](#) allow you to confine the view to a particular part of the available data. For example, when you move a Product field to the [filter area](#), you can display data for one product at a time.

[Moving row or column fields to a higher or lower level](#)

When a view has multiple row and column fields, the fields that are closest to the detail data are referred to as inner fields. The other fields are outer fields. You can switch the inner and outer fields.

Product	Sold By
Dairy	Davolio
	Dodsworth
	Suyama
	Total
Meat	Davolio
	Dodsworth

- 1 Outer row field
- 2 Inner row field
- 3 Outer field items
- 4 Items for the inner field are repeated for each item in the outer field.

[Adding and removing fields](#)

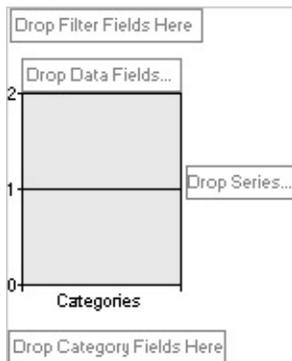
The layout of a PivotTable view does not have to include all of the fields that are available from the underlying [record source](#). You can also remove fields that you no longer want to see from the PivotTable view layout.

[Customizing the layout of a PivotChart view](#)

[Using the drop areas](#)

You change the layout of a chart by moving the fields to predefined drop areas within the chart workspace. To do this, drop areas must be shown in the chart.

Drop areas are displayed differently depending on the chart type. For example, a series drop area is not displayed for a pie chart because pie charts consist of only one series. If you plot multiple charts, you will see a drop area for multi-chart fields. This type of drop area is not displayed for single charts.



After you move fields to the drop areas so that the drop area captions are covered up, you can still drag additional fields to the areas.

[Moving fields to the category and series areas](#)

When you move a field to the series area, the unique items of data within the field are displayed as [data series](#) in the chart. These series are represented by colored [data markers](#), and their names appear in the chart [legend](#).

When you move a field to the category area, the unique items of data are displayed as categories, or related groups of data. Each category consists of one point from each data series. Category labels usually appear across the x axis of the chart, although this can vary depending on the type of chart you are using.

[Moving fields to the data area](#)

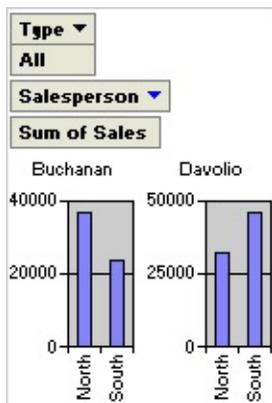
[Data fields](#) provide the values to be summarized in the chart. When you move a field to the data area, the values from the field are used as the data that is measured in the chart.

[Moving fields to the filter area](#)

A [filter field](#) is similar to a page field in a Microsoft Excel PivotTable report. Filter fields allow you to confine the view to a particular part of the available data. For example, when you move a Product field to the [filter area](#), you can have the chart display category and series values for one product at a time.

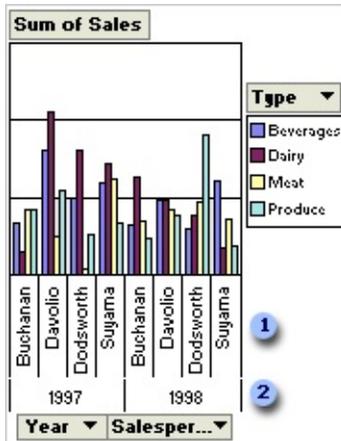
[Moving fields to the MultiChart area](#)

When you move a field to the MultiChart area, items in that field become separate charts. For example, if you move the Salesperson field to the MultiChart area, a chart is created based on data for each salesperson in that field. In the following example, the Salesperson field is in the MultiChart area, but it's filtered so it displays individual charts for Buchanan and Davolio.



[Moving the category or series fields to inner or outer levels](#)

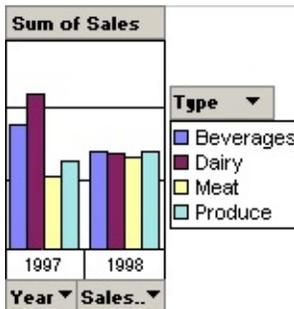
When a chart contains multiple [series](#) or [category fields](#), the fields that are closest to the data are referred to as inner fields. The other fields are outer fields. In the following example, Year is the outer field and Salesperson is the inner field. The inner field items are displayed as salespeople's names, and the outer field items are displayed as the years 1997 and 1998.



1 Inner field items

2 Outer field items

You can expand and collapse multiple fields to show more or less information in a particular field. For example, you can collapse the outer field (Year) in the example so that the inner field items are no longer displayed.



[Adding and removing fields](#)

The layout of a chart does not have to include all fields that are available from the source data. If more data is available, you can add fields to the chart. For example, if the chart summarizes sales revenue, and the source data also includes sales quantities, you might add the Quantity field as a [data field](#) to summarize both revenue and quantity of products sold. You can also remove fields from the chart layout that you no longer want to see.

[Impact of changing layout on filtering](#)

Filter settings are retained when you move or remove a field. This means that

when you move a [series](#) or [category field](#) to the [filter area](#) and back, previously hidden items are again hidden. If you remove a field and later add the field back to the layout, the same items are again hidden.

[Restricting use of a chart](#)

When you design a chart for other users, you can restrict the user's ability to change the layout of the chart by preventing fields from being added and moved.

[Show All](#)

View information about a field or item in PivotTable view

1. Open a datasheet or form in [PivotTable view](#).
2. Do one of the following:
 - For information about a field, click the label of the field.
 - For information about an [item](#) of data in a field, click the item.
3. On the **PivotTable** toolbar, click **Properties** , and then click the **Captions** tab.

[Show All](#)

Add, move, or remove a field in PivotTable or PivotChart view

[In PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Add a field to the row, column, filter, or detail area](#)

1. If the field list is not visible, click **Field List**  on the **PivotTable** toolbar.
2. Use the scroll bars and the expand indicators (+ and - boxes) to find the field you want to add to the view.
3. Click the field, and in the lower-right corner of the field list, do one of the following:
 - To add the field to the [row area](#), click **Row Area**.
 - To add the field to the [column area](#), click **Column Area**.
 - To add the field to the [filter area](#), click **Filter Area**.
 - To add the field to the [detail area](#), click **Detail Data**. To include only summarized values of a field, click **Data Area**.
4. Click **Add to**.

[Add a total field](#)

You can add [total fields](#) to the [detail area](#), but not to the [row](#), [column](#), or [filter area](#).

1. Open a datasheet or form in [PivotTable view](#).
2. Click the field that contains the values you want to summarize.
3. On the **PivotTable** toolbar, click **AutoCalc** , and then click the summary function you want.

[Add a calculated detail or total field](#)

You can add a [detail](#) or [total field](#) that uses an [expression](#) to calculate data.

1. Open a datasheet or form in [PivotTable view](#).
2. On the **PivotTable** toolbar, click **Calculated Totals and Fields** , and then do one of the following:
 - To create a detail field, click **Create Calculated Detail Field**.
 - To create a total field, click **Create Calculated Total**.
3. In the **Properties** dialog box, type the expression you want on the **Calculation** tab, and then click **Change**.

[Tip](#)

To add field references, select a field that you want to use from the **Insert Reference To** box, and then click the **Insert Reference To** button.

Notes

- To edit the expression that is being used in a calculated field, select the field, edit the expression, and then click **Change** on the **Calculation** tab of the **Properties** dialog box. The field values will be automatically recalculated using the new expression.
- By default, total fields that summarize numeric data use the **Sum** summary function. Fields that summarize other types of data use the **Count** summary function.

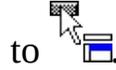
[Move a field to the row or column area](#)

1. Point to the field label of the field you want to move until the pointer changes to .
2. Do one of the following:
 - To make the field a [row field](#), click and drag the field label to the

[row area](#) on the left side of the view, until the pointer changes to



- To make the field a [column field](#), click and drag the field label to the [column area](#) near the top of the view, until the pointer changes



[Move a field to the detail area](#)

1. Point to the field label of the field you want to move until the pointer changes to
2. Click and drag the field to the [detail area](#) at the bottom right of the view, until the pointer changes to

[Move a field to the filter area](#)

1. Point to the field label of the field you want to move until the pointer changes to
2. Click and drag the field to the [filter area](#) at the top of the PivotTable view, until the pointer changes to

[Move a row or column field to a higher or lower level \(Change the column order of fields\)](#)

1. Point to the field label of the field you want to move to a different level until the pointer changes to
2. Do one of the following:
 - To move the field to a higher level, drag the field label toward the left and drop it where you want.
 - To move the field to a lower level, drag the field label toward the right and drop it where you want.

[Remove a field](#)

When you remove a field from the layout of a PivotTable view, data from

the field is no longer displayed but the field remains available in the field list for viewing.

1. Click the field label of the field that you want to remove from the view.
2. Press the DELETE key.

Note You can delete [total fields](#) and calculated [detail fields](#) from both the PivotTable view and the field list by right-clicking the field and then clicking **Delete** on the shortcut menu.

[In PivotChart view](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Do one or more of the following:

[Add a field to the series, category, data, MultiChart, or filter area](#)

1. On the **PivotChart** toolbar, click **Field List** .
2. Use the scroll bars and the expand indicators (+ and - boxes) to find the field you want to add to the chart.
3. Click the field and drag it to the appropriate drop area on the chart: the series area, category area, data area, MultiChart area, or filter area.

You can add multiple fields to each location.

Notes

- As an alternative to dragging the field, you can select the area to which you want to add a field in the list at the bottom of the field list, and click **Add to**. If **Add to** becomes unavailable when you select the area where you want to put the field, the field you selected might not be available for use as a category, series, data, or filter field.
- You can add more than one field to an area on the chart. The placement of fields on the chart determines the hierarchical order in which the data is displayed. Outer fields can be expanded or collapsed to show more or less information in inner fields.

- The MultiChart area is available only if you have clicked **Multiple Plots**  on the **PivotChart** toolbar.

[Change the summary function for a data field](#)

1. Click the [data field](#) that contains the values you want to summarize.
2. On the **PivotChart** toolbar, click **AutoCalc** , and then click the summary function you want.

[Move a field](#)

1. Open a datasheet or form in [PivotChart view](#).

In the view, click the field you want to move, and then drag it until the pointer changes to represent the area where you want to move it.

This pointer	Represents this area
	Series area
	Category area
	Filter area
	Data area
<input type="checkbox"/>	MultiChart area

2. Release the mouse button.

Note The pointers shown above apply only when you're moving fields within the chart, not when you're adding fields from the field list.

[Remove a field](#)

When you remove a field from a chart, data from the field is no longer displayed, but the field remains available in the field list for viewing.

1. Click the field label of the field that you want to remove from the chart.
2. Drag the field toward the outside of the chart until the pointer changes

to  .

[Show All](#)

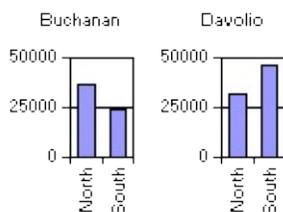
Change the chart type in PivotChart view

1. Open a datasheet or form in [PivotChart view](#).
2. Do one of the following:
 - To change the chart type for the entire chart, select the chart workspace by clicking the blank area outside the plot area but within the chart boundaries.
 - To change the chart type for an individual [data series](#), select the data series.
3. On the **PivotChart** toolbar, click **Chart Type** .
4. In the list, click the chart type you want, and then click the subtype you want.

[Show All](#)

Display multiple charts in PivotChart view

You can create multiple charts in the same [PivotChart view](#) to make it easy to compare different aspects of your data. The following example shows multiple charts that compare sales data by using individual charts for each salesperson. The charts in the example also use the same scale, which is optional.



1. Open a datasheet or form in PivotChart view.
2. On the **PivotChart** toolbar, click **Multiple Plots** .
3. If you want to use the same axis scale for all charts, click **Multiple Plots Unified Scale** .
4. On the **PivotChart** toolbar, click **Field List** , and then use the scroll bars and the expand indicators (+ and - boxes) to find the field you want to add to the chart.
5. Click the field and drag it to the [drop area](#) on the chart that says **Drop MultiChart Fields Here**. You can add multiple fields to the MultiChart area.
6. If you want to specify the layout of the multiple charts, click **Properties**  on the **PivotChart** toolbar, and then click the **General** tab.
7. Make sure **Chart Workspace** is selected in the **Select** box.
8. Under **Chart workspace**, click **Horizontal** or **Vertical** in the **Chart layout** list.

If you select **Automatic**, the charts will be displayed horizontally or

vertically depending on the size and shape of the workspace.

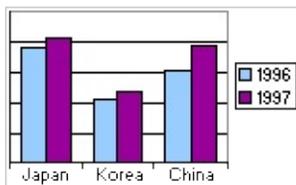
9. If you want to limit the number of charts across and down the chart workspace, type in the **Maximum charts per row/col** box the maximum number of charts you want displayed.

[Show All](#)

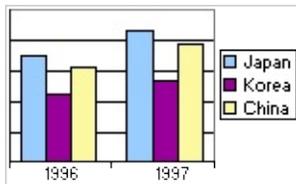
Change PivotChart view series to categories, or categories to series

1. Open a datasheet or form in [PivotChart view](#).
2. On the **PivotChart** toolbar, click **By Row/Column** .

Changing series to categories or categories to series alters the emphasis on which aspects of data you want to compare. For example, the following chart emphasizes the comparison of countries (as categories).



The following example emphasizes the comparison of years (as categories).



[Show All](#)

Show or hide elements of a PivotTable or PivotChart view

[In PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Show or hide drop areas](#)

When the [drop areas](#) are hidden and the PivotTable view does not have a field in the row, column, data, or filter area, you can't drag fields to these areas.

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Behavior** tab.
2. Select or clear the **Drop areas** check box.

[Show or hide ScreenTips](#)

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Report** tab.
2. Select or clear the **ScreenTips** check box.

[Show or hide totals](#)

1. Click the row or column field for which you want to show or hide subtotals and grand totals.
2. To show totals, on the **PivotTable** toolbar, click the **Subtotal** button  so that it is selected. To hide totals, click the button again.

Notes

- Subtotals and grand totals are not displayed for fields in the [detail area](#) (the area where all of the individual records can be displayed).
- When you choose to display subtotal and grand total values, they are displayed for every [total field](#) in the PivotTable view. You can't exclude individual total fields.
- Subtotal and grand total calculations can include or exclude items of data that are hidden by filtering.

[Show or hide details for an item or field](#)

Do one of the following:

- To show or hide detail data for a cell or for an item at the lowest level, click **Show Details**  or **Hide Details**  on the **PivotTable** toolbar.
- To show or hide detail data for all members of a row or column field, select the innermost row or column field, and then click **Show Details**  or **Hide Details**  on the **PivotTable** toolbar.

Note To show or hide lower-level items, select the row or column field or item, and then click **Expand**  or **Collapse**  on the **PivotTable** toolbar.

[Show or hide lower-level items of an item or field](#)

Do one of the following:

- To show or hide the lower-level items of a specific item in the row or column axis, click the expand indicator (+ or - box) next to the item.
- To show or hide the lower-level items for all items in a row or column field, click the field label, and then click **Expand**  or **Collapse**  on the **PivotTable** toolbar.

Notes

- If the + or - box is not visible next to the item, select the **Expand indicator** check box on the **Behavior** tab in the **Properties** dialog box. If after

selecting this check box you don't see an expand indicator next to an item, the item might be at the lowest level.

- To show or hide details, select the item or field, and then click **Show Details**  or **Hide Details**  on the **PivotTable** toolbar.

[Show or hide expand indicators next to items](#)

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Behavior** tab.
2. Select or clear the **Expand indicator** check box.

Notes

- This setting does not affect the  and  boxes that appear to the right of the items at the lowest level.
- Items at the lowest level in a row or [column area](#) don't have expand indicators on their left.

[Show or hide items that have no data](#)

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Report** tab.
2. Select or clear the **Empty rows** and **Empty columns** check boxes.

[In PivotChart view](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Do one or more of the following:

[Show or hide field buttons and drop areas](#)

1. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
2. Under **General commands**, in the **Select** box, click **Chart Workspace**.

3. Click the **Show/Hide** tab.
4. Select or clear the **Field buttons/drop zones** check box.

[Show or hide ScreenTips](#)

1. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
2. Under **General commands**, in the **Select** box, click **Chart Workspace**.
3. Click the **Show/Hide** tab.
4. Under **Show by default**, select or clear the **ScreenTips** check box.

[Show or hide the field list](#)

1. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
2. Under **General commands**, in the **Select** box, click **Chart Workspace**.
3. Click the **Show/Hide** tab.
4. Under **Show by default**, select or clear the **Field list** check box.

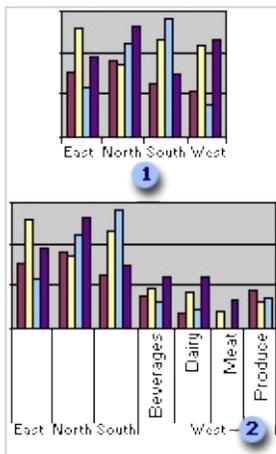
[Show or hide passive alerts](#)

A passive alert indicates that your chart contains an error. The alert appears as an exclamation point at the bottom of the chart, and when you click it, information about the error is displayed.

1. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
2. Under **General commands**, in the **Select** box, click **Chart Workspace**.
3. Click the **Show/Hide** tab.
4. Under **Errors**, select or clear the **Show passive alerts** check box.

[Show or hide details for an item or field by expanding or collapsing](#)

You can expand or collapse items in [series fields](#) or [category fields](#) to show or hide more detailed information for a particular item. Expanding is similar to drilling except that items on the same level are not hidden. In the example, West is expanded to show detailed information for West, but other regions are still displayed without detailed information. Collapsing will remove the detailed information for that item, and return the chart to the state it was prior to expanding.



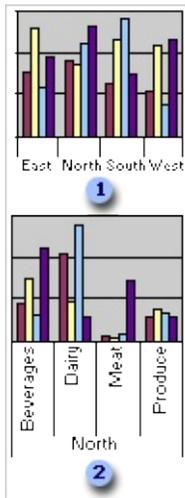
- 1 Chart that shows sales in each region
- 2 Chart with West expanded to show sales per product in the West region
 1. Click the field that contains the items you want to expand or collapse.
 2. On the **PivotChart** toolbar, click **Expand**  or **Collapse** .

[Show or hide details for an item or field by drilling](#)

You can drill into an item in an outer field to focus on detailed information for that particular item, or you can drill out to remove detailed information for that item. Drilling is similar to expanding, except that when you drill into an item, other items on the same level are hidden.

In the example, the region North is drilled into, showing detail information for the North region and hiding all information for other regions. When you drill out of an item that you have drilled into, the detailed information is removed, and

items on the same level are shown again.



1 Chart that shows sales information for products by region.

2 Chart that shows the North region item drilled into to show sales information for each type of product in the North region only. Other regions are hidden.

1. Click the item on the category axis for which you want to show or hide details.

2. On the **PivotChart** toolbar, click **Drill into**  or **Drill out** .

[Show All](#)

Add or change captions, labels, and titles in PivotTable or PivotChart view

[In PivotTable view](#)

The changes you make don't affect the source data.

1. Open a datasheet or form in [PivotTable view](#).
2. On the **PivotTable** toolbar, click **Properties** .
3. Do one or more of the following:

[Change the caption of a drop area](#)

1. In the **Properties** dialog box, click the **Captions** tab.
2. In the **Select caption** list, select the element whose caption you want to change.
3. In the **Caption** box, type the new caption.

[Change the caption of a field](#)

1. Click the field whose caption you want to change.
2. In the **Properties** dialog box, click the **Captions** tab.
3. In the **Select caption** list, select the element whose caption you want to change.
4. In the **Caption** box, type the new caption.

[Change the caption of an item or custom group](#)

1. Click the [item](#) or [custom group](#) whose caption you want to change.
2. In the **Properties** dialog box, click the **Captions** tab.
3. In the **Caption** box, type the new caption.

Notes

- If you don't want a caption for the selected item, clear the **Caption** text box.
- On the **Captions** tab, selecting **Report title bar** in the **Select caption** list and typing a new caption or applying a format does nothing.

[In PivotChart view](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Do one or more of the following:

[Add data labels](#)

The chart type associated with the selected [data series](#) or [data point](#) determines the type of data labels that can be added. You can change data labels after you add them.

1. Select the data series to which you want to add data labels by clicking an individual [data marker](#) in the series. Click again to select the entire series.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Click **Add Data Label** .

Note You cannot add a data label to a single data point. However, after adding data labels to a series, you can select and delete individual data labels that you do not want to display.

[Remove data labels](#)

1. Click the data label you want to remove.

If you want to remove all data labels in a [series](#), click a second time to select all data labels in the series.

2. Press DELETE.

[Change data labels](#)

[Change the contents of data labels](#)

1. Click a data label in the set of data labels you want to change, and then click a second time to select all data labels in the [series](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Data Labels** tab.
3. Under **Data Label**, select one or more of the following that you want shown in the data label: **Category name**, **Value**, **Series name**, **Bubble size** (available for bubble charts only), or **Percentage**.
4. If you select more than one of the above items to show in your data labels, you can separate each with a comma or space, or put each on a separate line. To do so, click the option you want in the **Separator** list.

[Change the text format of data labels](#)

1. Click a data label in the set of data labels you want to change, and then click a second time to select all data labels in the [series](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Format** tab.
3. Select the options you want under **Text format**.

[Change the position of data labels](#)

1. Click a data label in the set of data labels you want to change, and then click a second time to select all data labels in the [series](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Data Labels** tab.
3. Under **Data Label**, click the location in the **Position** list where you want the data labels to appear in relationship to the [data markers](#). For example, click **Inside Base** to show data labels inside and at the bottom of data

markers in a standard column chart.

[Change the border or fill for data labels](#)

1. Click a data label in the set of data labels you want to change, and then click a second time to select all data labels in the [series](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Border/Fill** tab.
3. Select the options you want under **Border** and under **Fill**.

[Add a title](#)

1. Click the item in the chart to which you want to add a title— for example, the chart workspace or an axis.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Click **Add Title** .

[Remove a title](#)

1. Click the title you want to remove.
2. Press DELETE.

[Change a title](#)

1. Click the title you want to change.
2. On the **PivotChart** toolbar, click **Properties** .
3. Do one or more of the following:

[Change the text format](#)

1. Click the **Format** tab.

2. Select the options you want under **Text format**.

[Change the title position](#)

1. Click the **Format** tab.
2. Under **Position**, in the **Position** box, click **Top**, **Bottom**, **Left**, or **Right**.

[Change the title contents](#)

1. Click the **Format** tab.
2. Under **Caption**, type a new title in the **Caption** box.

[Change the border or fill color](#)

1. Click the **Border/Fill** tab.
2. Select the options you want under **Border** and under **Fill**.

[Show All](#)

About formatting a PivotTable or PivotChart view

[Formatting a PivotTable view](#)

You can format [row](#), [column](#), [total](#), or [detail fields](#), and [drop area](#) captions in the following ways:

- **Setting text formats** You can set the font, font size, and color for text, and make text bold, underlined, or italic.
- **Aligning text in a cell** You can display values so that they are left-aligned, right-aligned, or centered horizontally in the cells. You cannot change the values' vertical alignment.
- **Setting background colors** You can change the background color for cells.
- **Changing number formats** You can change the formats in which numbers, dates, and times in a field are displayed to any of several predefined number formats. For example, you can display a number as currency or as a percentage, and you can display a date in long format, with the day and month spelled out, or in abbreviated format. Changes to the number format do not change the actual values, only how the values are displayed.

Formats that you apply to one element might affect similar elements. For example, if you change the text color for an item, all other items in the same field are formatted with the new text color.

[Formatting a PivotChart view](#)

Following are some of the ways you can format items in a PivotChart view.

Setting text formats You can set the font, font size, and color for text in titles, labels, and in the [legend](#), and make text bold, underlined, or italic.

Changing number formats You can change the formats in which numbers, dates, and times in a field are displayed to any of several predefined number formats. For example, you can display a number as currency or as a percentage, and you can display a date in long format, with the day and month spelled out, or in abbreviated format. Changes to the number format do not change the actual values, only how the values are displayed.

Changing colors You can change background and border colors for the entire chart, the plot area, a [data series](#) or individual [data markers](#), or for the legend in the chart.

Changing line color, weight, and style You can change formats of gridlines, axis lines, border lines, and more.

Adding pictures, textures, and fills You can add pictures or fills such as gradients to items in the chart such as the plot area, data markers, and more.

Viewing 3-D charts You can view 3-D charts in a variety of different perspectives and projections, and change the lighting source for different effects.

Mapping chart colors to data You can format data markers so that graduated or solid colors represent values or ranges of data. For example, you can show sales amounts for each salesperson in data markers that are distinguished by graduated colors, where higher sales amounts are represented by lighter or darker colors.

Adding trendlines or error bars You can add trendlines to data series to show trends in data or to forecast future values. You can add error bars to data series to show potential error amounts for data.

Changing axes You can format axes in charts to measure data based on different scales, including timescale axes. You can add and modify formatting for axis titles, labels, and more.

Note Formats that you apply to a datasheet in [PivotTable view](#) or [PivotChart view](#) are not replicated when you set up replication for your database. Formats that you apply to a form in these views will be replicated.

[Show All](#)

Format an element in PivotTable or PivotChart view

[In PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Select the element that you want to format.
3. On the **PivotTable** toolbar, click **Properties** .
4. Do one of the following:
 - To format a [drop area](#), click the **Captions** tab.
 - To format all other elements of a PivotTable view, click the **Format** tab.
5. Do one or more of the following:

[Specify font and style settings](#)

- To make the text bold, click **Bold** .
- To make the text italic, click **Italic** .
- To underline the text, click **Underline** .
- To change the text color, click the arrow next to **Font Color** , and then click the color you want.
- To change the font, click a different font in the **Font** list.
- To change the font size, select a size from the list next to the **Font** list.

Note The fonts and font sizes available in the lists are determined by the fonts you have installed on your computer.

[Align text](#)

- Click the alignment you want: **Align Left** , **Align Center** , or

Align Right .

Note You cannot change the horizontal alignment of row, column, or filter field labels.

[Set background color](#)

- Click the arrow next to **Fill Color** , and then click the color you want.

Note You cannot set the background color of an entire PivotTable view in one operation. You cannot change the background color of the [drop areas](#).

[Change the number format of a field, total, or custom property of a row or column field](#)

- In the **Number** list, click the format you want.

Note The number format of a field does not affect the number format of other row, column, filter, or total fields.

[In PivotChart View](#)

[Rotate or flip the plot area](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the plot area that you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
4. Do one of the following:
 - To flip the plot area horizontally or vertically, click **Flip Horizontal**  or **Flip Vertical** .
 - To rotate the plot area clockwise or counterclockwise in 90-degree increments, click **Rotate Clockwise**  or **Rotate Counterclockwise**



[Change the text size and font for chart items](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the axis title, data labels, R-squared label and equation label, legend text, chart title, or other chart text that you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Format** tab.
4. Under **Text format**, select the options you want.

[Change borders and background colors](#)

You can change the border weight and color and the background color for the chart workspace, the plot area, the [legend](#), a single [data marker](#), or a series of data markers.

1. Open a datasheet or form in [PivotChart view](#).
2. Select the chart area or item you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Border/Fill** tab.
4. Do one of the following:
 - To change the border color, weight, or style, select the options you want under **Border**.
 - To change the interior color or fill, select an option under **Fill Type** (**Solid color**, **Pattern**, **Gradient**, or **Picture/Texture**) and then select the options you want that pertain to that fill type. For example, if you select **Gradient**, you can then choose the number of colors in the gradient and the gradient style, or you can choose a preset gradient.

[Change line color, weight, and style in chart items](#)

You can change the line weight and color of [series lines](#), [axis lines](#), gridlines, [trendlines](#), error bars, and other types of lines in a chart.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the line you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Line/Marker** tab.
4. Under **Line**, select the options you want.

[Show, hide, or change gridlines](#)

[Show or hide gridlines](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the axis for which you want to show or hide gridlines.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Axis** tab.
4. Under **Ticks and Gridlines**, select or clear either or both of the **Major gridlines** and **Minor gridlines** check boxes.

[Change gridlines](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the gridlines you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Line/Marker** tab.
4. Under **Line**, select the options you want.

[Add a picture or textured fill to a chart item](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Select the chart area or item to which you want to add a picture— for example, the chart workspace, the plot area, or a [data marker](#).
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Border/Fill** tab.
4. Under **Fill**, in the **Fill Type** list, click **Picture/Texture**.
5. Do one of the following:

[Add your own picture](#)

1. Click **URL** and type the location and name of the image you want to use.

For location, you can use a Web address starting with a valid protocol such as:

`http://www.microsoft.com/images/image_name.gif`

or a UNC path such as:

`\\server\folder\image_name.gif`

or a path to a file on your hard drive, such as:

`C:\image_image.gif`

2. Press ENTER.
3. Next to **Format**, click an option for how you want the picture to appear in the chart.
 - **Tile** Tiles the image to fill the selected space.
 - **Stretch** Stretches the image to fill the selected space.
 - **Stack** Stacks the image to fill the selected space.

- **Stack scale** Stacks the image according to the number of value axis units you specify in the **units** box. For example, if a data marker in a column chart measures \$6,000 on the value axis, and you specify units to represent \$1,000, six images would be stacked in the data marker.
 - **Stretch plot** Stretches the image across the entire chart, but fills only the selected item or items.
4. If you chose **Stack scale**, type the units you want each image to fill in the **units** box, and press ENTER.

[Add a built-in texture](#)

- Click **Preset** and click the texture you want to use in the **Preset** list.

[Change the view of a 3-D chart](#)

1. Open a datasheet or form in [PivotChart view](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Under **General commands**, click **Chart Workspace** in the **Select** box.
4. Click the **3D View** tab.
5. Click the **Projection mode** you want.

Perspective  Chart objects are represented in relative distance and positions as they might appear to the eye.

Orthographic  Chart objects are represented as if they all appear on the same plane, on a surface that is perpendicular to both the view and the lines of projection.

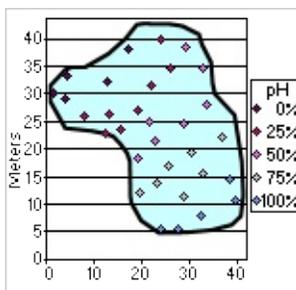
6. Make the changes you want by dragging the sliders for each option. For example, to rotate the lighting source for the chart, drag the the rotation slider under **Lighting**.

Notes

- To return to the default view of the chart, click **Default**  next to **Default view**.
- To make the height ratio or depth ratio proportionate, select the **Height Ratio** or **Depth Ratio** check box.

[Map chart colors to distinguish ranges of data](#)

You can format [data markers](#) so that graduated or solid colors represent values or ranges of data. For example, the following chart shows locations of water samples and corresponding pH measurements distinguished by graduated colors, where higher percentages of pH levels are represented by increasingly lighter colors.



1. Open a datasheet or form in [PivotChart view](#).
2. Click a data marker within the [data series](#) you want to format; then click again to select the entire data series.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Conditional Format** tab.
4. Select the **Conditionally color data points** check box.
5. In the **Style** list, click the style of conditional formatting you want.
 - **Three Point Gradient** Uses three graduated colors to distinguish between the smallest value and zero (or the value you indicate) and between zero (or the value you indicate) and the largest value.

- **Two Point Gradient** Uses two graduated colors to distinguish between smallest and largest values.
 - **Two Segment Solid** Uses two solid colors to distinguish between values smaller than zero (or the value you indicate) and larger than zero (or the value you indicate).
 - **Bottom Values** Uses a solid color to distinguish the smallest values in a range.
 - **Top Values** Uses a solid color to distinguish the largest values in a range.
6. Next to **Colors**, choose the **Beginning Color**, **Middle Color**, or **End Color** (where applicable) to represent value ranges.
7. If you want to indicate a value on which to base your conditional format, type that value in the **Value** box and select the **%** check box if appropriate. For example, to format the top 90% of values, type **90** in the value box and then select the **%** check box.
- You can specify additional values on which to base conditional formatting. For example, if you plot sales data for products in different regions, you can format the data markers based on an additional data set such as profit amount. To do this, clear the **Conditional format based on data value field** check box, and then add the [data field](#) you want to use to the **Format Data** [drop area](#) that appears on the chart.

[Pull out slices in pie or doughnut charts](#)

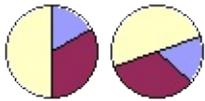


In doughnut and stacked pie charts, you can pull out slices in the outer ring only.

1. Open a datasheet or form in [PivotChart view](#).
2. Do one of the following:

- To pull out all slices or rings, click the chart workspace (the blank area between the plot area and the chart boundary).
 - To pull out one slice or ring, click the data slice or ring.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
 4. Under **Chart**, in the **Explosion** box, type a number to represent— as a percentage of the radius— the distance between the point of the slice or slices and the center of the doughnut or pie.

[Rotate the position of the slices in a doughnut or pie chart](#)



The order in which [data series](#) are plotted in pie and doughnut charts is determined by the order of the source data. Use this procedure to rotate slices within the 360 degrees of the pie or doughnut circle.

1. Open a datasheet or form in [PivotChart view](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. In the **Select** list, click **Chart Workspace**.
4. Under **Chart**, in the **Angle of first slice** box, type a number between 0 and 360 to specify the angle at which you want the first slice to appear.

[Change the overlap or spacing of data markers in bar and column charts](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the chart workspace (the blank area between the plot area and the chart boundary).
3. On the **PivotChart** toolbar, click **Properties** , and then click the **General**

tab.

4. Do one of the following:

- To change the overlap of the [data markers](#) in each category, type a number between -100 and 100 in the **Overlap** box to represent the percentage of overlap you want. For example, at 100, bars or columns overlap each other, while at -100, a bar or column width of space is displayed between each. At 0, bars or columns are adjacent to each other.
- To change the spacing between categories of data markers, type a number between 0 and 500 in the **Gap width** box to represent the percentage of gap width you want. For example, at 0, data markers touch; at 100, there is one bar or column width of space between each.

[Show All](#)

Format the contents of a PivotTable view to read right to left or left to right

When you select right-to-left display, the elements of the [PivotTable view](#) are displayed as a mirror image of the default left-to-right display. For example, the drop areas are on the upper-right side of the view, [row fields](#) are in the rightmost columns, and items in [column fields](#) are listed from right to left.

1. Open a datasheet or form in PivotTable view.
2. On the **PivotTable** toolbar, click **Properties** , and then click the **Behavior** tab.
3. Select or clear the **Display right to left** check box.

Note If the form or datasheet is set for right-to-left display, the PivotTable view will automatically be set for right-to-left display.

[Show All](#)

Add, remove, or change axes in PivotChart view

[Add an axis](#)

1. Open a datasheet or form in [PivotChart view](#).

Click the chart workspace (the blank area between the plot area and the chart boundary).

2. On the **PivotChart** toolbar, click **Properties** , and then click the **Series Groups** tab.
3. Under **Series groups**, in the **Select one or more series** box, select the series group for which you want to add an axis.
4. If you want the axis to reflect a different series, you can group, ungroup, or merge series with other groups.
5. Under **Add axis**, in the **Groups** box, select the group that you want to add the axis for, and then in the **Axis position** box, select where you want the axis to be displayed.
6. Click **Add**.

[Remove an axis](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the axis you want to remove.
3. Press DELETE.

[Change the position of an axis](#)

1. Open a datasheet or form in [PivotChart view](#).

2. Click the axis you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Format** tab.
4. Under **Position**, in the **Position** box, select the option you want.

[Change the spacing of tick marks and labels on the category axis](#)

Each category on the category axis is identified by a label and separated from other categories by tick marks. You can change the intervals at which labels and tick marks appear, and you can specify where you want the value (y) axis to cross the category (x) axis.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the category axis you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Scale** tab.
4. Under **Scale**, do one or more of the following:
 - Type a number in the **Labels spacing** box to specify the frequency at which category labels must be labeled. For example, type **1** to label every category, type **2** to label every other category, and so on.
 - Type a number in the **Tick mark spacing** box to specify the number of categories you want between each pair of tick marks. For example, type **1** to have one category between each tick mark; or type **2** to have two categories between each tick mark.

[Change where the category \(x\) axis crosses the value \(y\) axis](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the category axis you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Scale** tab.
4. In the **Crosses with** list, under **Crossing Axis**, click the axis that you want to cross the category axis.

5. If you want to specify where the crossing axis should cross the category axis, select the **Custom** check box and then type a value in the box next to it.

[Change a timescale category axis](#)

When your category data is made up of numeric dates, the chart automatically uses a timescale axis. The timescale category axis displays dates in chronological order at specific intervals even if the dates in your data are not in order.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the category axis you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Scale** tab.
4. Under **Timescale**, click an option in the **Timescale** box:

None Rather than a timescale axis, use a category axis that uses your dates as text labels.

Auto The chart uses a timescale axis and determines the most appropriate **Group size** (that is, whether to group dates by days, months, quarters, and so on) for the axis based on your data.

Manual The chart uses a timescale axis, but you determine the most appropriate **Group size** (that is, whether to group dates by days, months, quarters, and so on) for the axis based on your data.

5. If you chose **Auto** or **Manual** in the previous step, select how you want dates displayed and totaled in the remaining options under **Timescale**.

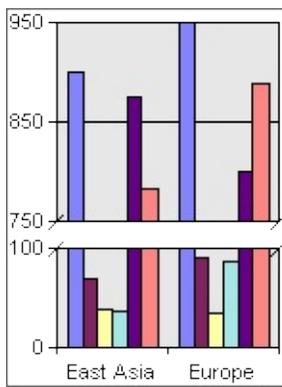
[Reverse the direction of values on an axis](#)

1. Open a datasheet or form in [PivotChart view](#).
2. Click the axis for which you want to reverse the direction of values.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Scale** tab.

4. Under **Order**, select the **Show values in reverse order** check box.

[Split values in an axis scale](#)

When you have very low and very high values in your chart, you can split the value axis (or timescale axis) so that all of the values are shown clearly. For example, if your values range from 0 to 950, with values falling between 0 and 100 and between 750 and 950, you can split the axis so that it has two scales: The bottom scale ranges from 0 to 100, and the top scale ranges from 750 to 950, thus eliminating unnecessary chart space between 100 and 750.



1. Open a datasheet or form in [PivotChart view](#).
2. Click the value axis that you want to split.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Axis** tab.
4. Select the **Split axis** check box, type a value in the **Split begins at** box (which marks the end of the first scale), and then type a value in the **Split ends at** box (which marks the beginning of the second scale).

[Change the scale of a value \(y\) axis](#)

The scale specifies the range of values on an axis, the way the axis values are displayed, the intervals at which the values occur, and the point at which one axis crosses another.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the value axis you want to change.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Scale** tab.

4. Do one or more of the following:
- Under **Crossing Axis**, select in the **Crosses with** box the axis you want to cross the value axis. If you want to specify where the crossing axis should cross the value axis, click **Custom**, and then type a value in the **Custom** box.
 - To specify either or both the highest and lowest values you want to appear on the axis, select the **Custom max** or **Custom min** check box, or both, and then type the appropriate values in the adjacent boxes.
 - If you want to specify the intervals between major or minor tick marks and gridlines, select the **Custom major unit** or **Custom minor unit** check box, or both, and enter the appropriate values in the adjacent boxes to specify the intervals you want.
 - To recalculate values on the axis based on a power of 10, 2, or e, select the **Logarithmic scale** check box, and then click the power you want to use in the **Use base** list.

[Show All](#)

Add, remove, or change error bars in PivotChart view

Error bars graphically express potential error amounts relative to each [data marker](#) in a [data series](#). You can add one set of error bars per series.

1. Open a datasheet or form in [PivotChart view](#).
2. Do one of the following:

[Add error bars](#)

1. Click a data marker in the data series to which you want to add error bars, and then click again to select the entire series.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Under **Add**, click **Add Errorbar** .

[Remove error bars](#)

You cannot delete an individual error bar. When you delete one error bar, all error bars for the entire data series are deleted.

1. Click any error bar in the series for which you want to delete error bars.
2. Press DELETE.

[Change error bar settings](#)

When you select an error bar, any changes you make affect all the error bars in its associated data series.

1. Click any error bar in the series for which you want to change settings.

2. On the **PivotChart** toolbar, click **Properties** , and then click the **Error Bars** tab.

3. Do one or more of the following:

- To select a different type of error bar (plus or minus), click the arrow next to **Display** , and then click the type you want.
- To show or hide end caps on the error bars, click the arrow next to **Marker** , and then click the type you want.
- To change the method used to determine the error amount, click the method you want in the **Error calculation** list, and then type the number you want in the **Error** box or the **Percentage** box.
- To change the direction of the error bars so that they calculate based on x-values or y-values for xy (scatter) charts, polar charts, and bubble charts, click **X** or **Y** in the **Direction** list.

[Show All](#)

Add, remove, or change a trendline in PivotChart view

1. Open a datasheet or form in [PivotChart view](#).
2. Do one or more of the following:

[Add a trendline](#)

1. Click a [data marker](#) within the [data series](#) to which you want to add a trendline, and then click again to select the entire series.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Under **Add**, click **Add Trendline** .

Note A linear trendline, its equation, and its R-squared value are placed on the series. Other trendline types include logarithmic, polynomial, power, exponential, and moving average.

[Remove a trendline](#)

1. Click the trendline you want to delete.
2. Press DELETE.

[Change a trendline](#)

1. Click the trendline you want to change.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Trendline** tab.
3. Do one or more of the following:
 - To change the trendline type, click the arrow next to the **Type** box and

select a type in the list.

- To change the order of a polynomial trendline, click the arrow next to the **Order** box and select from the list the number you want to be the highest power for the independent variable.
- To change the trendline name, which appears in the [legend](#), click the **Custom Name** check box under **Name**, and then type a new name in the box to the right.
- To show or hide the equation you use to calculate the trendline, select or clear the **Display equation** check box.
- To show or hide the R-squared value, select or clear the **Display R-squared value** check box.

[Show All](#)

Add, remove, or change a legend in PivotChart view

1. Open a datasheet or form in [PivotChart view](#).
2. Do one or more of the following:

[Add or remove a legend](#)

1. Select the chart workspace by clicking the blank area outside the plot area but within the chart boundaries.
2. On the **PivotChart** toolbar, click **Show Legend** .

[Change a legend](#)

1. Click the [legend](#) you want to change.
2. On the **PivotChart** toolbar, click **Properties** .
3. Do one of the following:

[Change the legend position](#)

1. Click the **Format** tab.
2. Under **Position**, in the **Position** box, click **Top**, **Bottom**, **Left**, or **Right**.

[Change the border weight or color and background color](#)

1. Click the **Border/Fill** tab.
2. Under **Border**, select the **Color**, **Weight**, or **Dash** options you want.
3. Under **Fill**, select the **Fill Type** and **Color** you want.

[Change the font or size of text](#)

1. Click the **Format** tab.
2. Under **Text format**, select the options you want.

[Show All](#)

Number formats available in PivotTable or PivotChart view

The regional settings in Microsoft Windows Control Panel determine how number, currency, date, and time formats are displayed. For example, these settings determine whether years are displayed with two digits or four digits. See Windows Help for information about these formats.

The following table lists the number formats supported in [PivotTable view](#) and [PivotChart view](#).

Format	Effects
	Displays a number right-aligned, with no thousands separator, and preceded by a minus sign if negative.
General	Displays up to 11 digits, or 10 digits with a decimal point, rounding additional digits to the right of the decimal point. Displays numbers with more than 11 digits to the left of the decimal point in scientific format.
	Displays text left-aligned.
	Displays a number as a date and optional time. For example, 8/31/99 05:54 AM.
General Date	The fractional part of the number is the time. If a number has no fractional part, only a date appears. If a number has only a fractional part and no integer part, only a time appears.
Long Date	Displays a date according to the Microsoft Windows regional setting for the long date format. For example, Thursday, January 25, 2001.
Medium Date	Displays a date using the abbreviated month names and hyphens (-) to separate month, day, and year. For example, Jan-25-2001.
Short Date	Displays a date according to the Windows regional setting for the short date format. For example, 1/25/2001.

Long Time	Displays a time according to the Windows regional setting for the long time format. For example, 8:45:36 PM.
Medium Time	Displays a time in 12-hour format with an AM or PM designator, omitting the seconds. For example, 8:45 PM.
Short Time	Displays a time in 24-hour format, omitting the seconds. For example, 20:45.
Currency	Displays a number according to the Windows regional setting for currency. For example, with the default United States English regional settings, \$2,532.75.
Euro Currency	Displays a number as euro currency, with a thousands separator and two decimal places. The placement of the euro sign and the decimal places are based on the regional settings for currency in Windows Control Panel. For example, with the default United States English regional settings, €12.35.
Fixed	Displays a number with two decimal places. For example, 68.30.
Standard	Displays a number according to the Windows regional settings for numbers. For example, with a comma as the thousands separator and a period as the decimal symbol, 1,800.00.
Percent	Displays a number multiplied by 100, with two decimal places and a percent sign. For example, the value 0.8914 would be displayed as 89.14%.
Scientific	Displays a number in exponential notation. For example, 1.25E+10.
Yes/No	Displays No if the number is 0; displays Yes for any other number.
True/False	Displays False if the number is 0; displays True for any other number.
On/Off	Displays Off if the number is 0; displays On for any other number.

[Show All](#)

Allow multiple selection in a filter field

You can specify whether a field in the [filter area](#) will allow single or multiple selection. In single-selection mode, selecting an [item](#) automatically clears the previous selection. In multiple-selection mode, when you select an item, previously selected items remain selected.

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. Select the field in the filter area for which you want to change the setting.
3. On the **PivotTable** or **PivotChart** toolbar, click **Properties** , and then click the **Filter and Group** tab.
4. Select or clear the **Allow selecting multiple items when in filter area** check box.

Notes

- By default, multiple selection is enabled.
- This setting applies only when the field is in the filter area. If you move the field to another area in the view, the field will always support multiple selection.

[Show All](#)

Automatically show items or details in PivotTable view

1. Open a datasheet or form in [PivotTable view](#).
2. On the **PivotTable** toolbar, click **Properties** , and then click the **Behavior** tab.
3. Do one or both of the following:

[Automatically show items](#)

- Set the **Expand items by default** option to **Always**.

[Automatically show details](#)

- Set the **Expand details by default** option to **Always**.

Note To hide all [items](#), set the **Expand items by default** option to **Never**. To hide detail data for all items, set the **Expand details by default** option to **Never**.

[Show All](#)

Prevent users from making changes in PivotChart view

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. On the **PivotChart** toolbar, click **Properties** , and then click the **Show/Hide** tab.
3. Under **Let users view**, clear one or more of the following check boxes:
 - **Allow pivot grouping**
 - **Allow pivot filtering**
 - **Selection marks**
 - **Properties dialog box**

Warning If you clear this check box, you will not be able to view the **Properties** dialog box again unless you close the PivotChart view without saving changes and then reopen it.

[Show All](#)

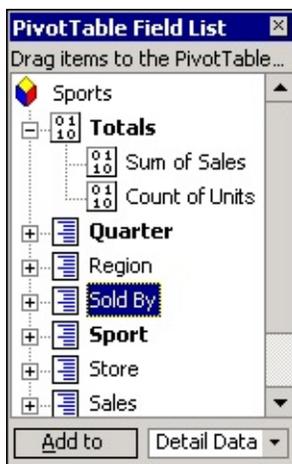
About calculating totals in PivotTable view

In [PivotTable view](#), you can summarize data in two ways.

[Total fields](#)

You create [total fields](#) to summarize data from other fields in the source data. For example, if the datasheet or form has a Sales field that records the amount of each individual sale, a Sum of Sales total field could add up the sales figures for each category, or a Count of Sales total field could tally the number of sales for each category. You can also create a total field that uses an [expression](#).

In the field list, the total fields are displayed at the top, under Totals. The Sum of Sales and Count of Units fields are available for inclusion in the PivotTable view as total fields.



For example, in the following PivotTable view, the Sum of Sales total field has been added to the [detail area](#) to display summary data for each sport and quarter.



Sport	Quarter	Sum of Sales
+	Golf	\$9,930
-	Tennis	
	Qtr3	\$4,670
	Qtr4	\$6,500

- 1 The Sum of Sales total field.

You cannot add total fields to the [row](#), [column](#), or [filter area](#).

A total field uses a summary function, such as **Sum** or **Count**. Depending on the type of source data, you can also use more than one summary function to summarize the data in the same field by adding more total fields, each using a different summary function.

You can also add a calculated total field that uses an expression. For example, to see the year-to-date sales figures, you could create a total field that uses the following expression:

Sum (YTD(), [Measures].[Sales])

The following PivotTable view shows sales and year-to-date sales figures.

Drop Filter Fields Here		Drop Column Fields Here	
Year	Qtr	Sales	YTD Sales
1997	Q1	\$1,396.35	\$1,396.35
	Q2	\$1,326.27	\$2,722.62
	Q3	\$1,402.89	\$4,125.51
	Q4	\$1,526.62	\$5,652.13
	Total	\$5,652.13	\$5,652.13

[Changing the orientation of total fields](#)

You can change the orientation of total fields in the PivotTable view. Displaying the fields in the row orientation lists them down the first column to the right of the [row fields](#). Displaying them in the column orientation lists them across the first row below the [column fields](#).

[Showing total values as percentages](#)

You can also show total values as a percentage of a row, column, or grand total, or as a total of the [item](#)'s parent in the row or column axis.

Drop Filter Fields Here				
		Employee ID		Grand Total
		1	3	
Exp Rpt Nm	Exp Type	Sum of Adv	Sum of Adv	Sum of Adv
Mktg Expo	100	78	890	968
	Total	78	890	968
Sales Conf	100	100		100
	200	3009		3009
	Total	3109		3109
Seminar	300	87	89	176
	Total	87	89	176
Grand Total		3274	979	4253

- 1 These values will be used if you choose to show total values as a percentage of row total.
- 2 These values will be used if you choose to show total values as a percentage of column total.
- 3 These values will be used if you choose to show total values as a percentage of the total of the item's parent in the row axis.
- 4 This value will be used if you choose to show total values as a percentage of the grand total.

Subtotals and grand totals

You can display subtotals for any row or column field in the PivotTable view if the view includes at least one [total field](#) and more than one [row field](#) or more than one [column field](#). If you choose to display subtotals for a field, grand totals are also selected for display, and subtotals and grand totals are displayed for every total field in the PivotTable view where the field intersects the row or column field you select.

In the following example, subtotals and grand totals are displayed for both row fields using the values in the Sum of Sales total field.

Sales Report		
Sport	Quarter	Sum of Sales
+	Golf	\$9,930
-	Tennis	
	Qtr3	\$4,670
	Qtr4	\$6,500
	Total	\$11,170
+	Grand Total	\$21,100

- 1 Grand total

2 Subtotal

When subtotals and grand totals are displayed for the Sport and Quarter fields, the PivotTable view includes subtotals for the Quarter field and a grand total for the Sport field.

If you move a field for which subtotals or grand totals are displayed out of the row or column area, the subtotals or grand totals become hidden, but the setting is retained. The subtotals or grand totals are redisplayed if you move the field back to the [row](#) or [column area](#).

When you hide subtotals and grand totals for a field, they remain hidden regardless of where you move the field, even if you remove the field from the PivotTable view and later add it back.

Depending on the type of data you're viewing in the PivotTable view, you might want subtotals and grand totals to include or omit the values for items of data you've hidden by filtering. You can specify that all subtotals and grand totals either include or exclude any hidden data.

[Show All](#)

Customize a total field in PivotTable view

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Change the orientation of total fields](#)

This procedure affects [total fields](#) displayed in the view. It has no effect on subtotals or grand totals.

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Report** tab.
2. Under **Display totals as**, select one of the following:
 - **Column headings** To display total fields across the first row below the [column fields](#) or in the top row if there are no column fields.
 - **Row headings** To display total fields down the first column to the right of the [row fields](#) or in the leftmost column if there are no row fields.

[Show totals as percentages](#)

- Click the [total field](#) whose values you want to view as percentages, click **Show As**  on the toolbar, and then click one of the following:
 - **Percent of Row Total** To display the values as a percentage of the total value of each row.
 - **Percent of Column Total** To display the values as a percentage of the total value of each column.
 - **Percent of Parent Row Item** To display the values as a percentage of the total value of the item's parent in the row axis.
 - **Percent of Parent Column Item** To display the values as a percentage of the total value of the item's parent in the column axis.
 - **Percent of Grand Total** To display the values as a percentage

of the PivotTable view's grand total value.

[Include or exclude hidden data in totals](#)

This procedure affects all subtotals and grand totals displayed in the view.

1. On the **PivotTable** toolbar, click **Properties** , and then click the **Report** tab.
2. Under **Calculate totals based on**, do one of the following:
 - To include hidden data, select **All items (including hidden items)**.
 - To exclude hidden data, select **Visible items only**.

Notes

- This setting affects data hidden by filtering [row](#) and [column fields](#); data hidden by using [filter fields](#) is always excluded from totals.
- When you choose to include hidden data, asterisks (*) appear next to the captions of subtotals and grand totals to note that the total values include hidden items as well as displayed items.

[Show All](#)

Functions available in PivotTable or PivotChart view of a form

A [control](#) in a [data access page](#) or a form that is open in [PivotTable view](#) or [PivotChart view](#) can contain an [expression](#) that uses a Microsoft Visual Basic for Applications (VBA) function. However, the expression will generate an error unless it appears on the following list.

Note This limitation doesn't apply to expressions in tables, queries, views, stored procedures, or user-defined functions that you have opened in PivotTable or PivotChart view.

[A - C](#)

ABS	ARRAY
ASC	ASCB
ASCW	ATN
CBOOL	CBYTE
CCUR	CDATE
CDBL	CHOOSE
CHR	CHR\$
CHRB	CHRB\$
CHRW	CHRW\$
CINT	CLNG
COS	CSNG
CSTR	CVAR
CVDATE	CVERR

[D - H](#)

DATE	DATE\$
DATEADD	DATEDIFF

DATEPART DATESERIAL
DATEVALUE DAY
DDB ERROR
ERROR\$ EXP
FIX FORMAT
FORMAT\$ FV
HEX HEX\$
HOUR

[I - L](#)

IIF IMESTATUS
INSTR INT
IPMT IRR
ISDATE ISEMPY
ISERROR ISNULL
ISNUMERIC ISOBJECT
ICASE ICASE\$
LEFT LEFT\$
LEFTB LEFTB\$
LEN LENB
LOG LTRIM
LTRIM\$

[M - Q](#)

MID MID\$
MIDB MIDB\$
MINUTE MIRR
MONTH NOW
NPER NPV
OCT OCT\$
PARTITION PMT
PPMT PV

QBCOLOR

[R - S](#)

RATE	RGB
RIGHT	RIGHT\$
RIGHTB	RIGHTB\$
RND	ROUND
RTRIM	RTRIM\$
SECOND	SGN
SIN	SLN
SPACE	SPACE\$
SQR	STR
STR\$	STRCOMP
STRCONV	STRING
STRING\$	SWITCH
SYD	

[T - Z](#)

TAN	TIME
TIMES\$	TIMER
TIMESERIAL	TIMEVALUE
TRIM	TRIM\$
TYPENAME	UCASE
UCASE\$	VAL
VARTYPE	WEEKDAY
YEAR	

[Show All](#)

Summary functions available in PivotTable or PivotChart view

The following table lists the summary functions provided by the **AutoCalc**  command that you use to create [total fields](#).

Function	Description
Sum	Calculates the sum of the values.
Count	Tallies the number of values. Empty values, such as zero-length strings (strings that contain no characters), are included in the count, but null values (values that indicate missing or unknown data) are not included.
Min	Returns the smallest value in the field being summarized.
Max	Returns the largest value in the field being summarized.
Average	Returns the average of the values in the field being summarized.
Standard Deviation	Returns the standard deviation of the values in the field being summarized, assuming the values are a sample of a larger population.
Standard Deviation Population	Returns the standard deviation of the values in the field being summarized, assuming the values are the entire population.
Variance	Returns the variance of the values in the field being summarized, assuming the values are a sample of a larger population.
Variance Population	Returns the variance of the values in the field being summarized, assuming the values are the entire population.

Note The summary functions available for a field depend on the type of data in the field. For example, if a field contains text, you cannot use **Sum**, **Min**, or **Max**, but you can use **Count** for that field.

[Show All](#)

Edit a calculated field in PivotTable view

1. Open a datasheet or form in [PivotTable view](#).
2. Select the calculated field whose [expression](#) you want to edit.
3. On the **PivotTable** toolbar, click **Properties** , and then click the **Calculation** tab.
4. Edit the expression, and then click **Change**.

The data values will be automatically recalculated using the new expression.



To add field references, select a field that you want to use from the **Insert Reference To** box, and then click the **Insert Reference To** button.

[Show All](#)

Show data as hyperlinks in PivotTable view

You can choose to display data as [hyperlinks](#) in a [row](#), [column](#), or [detail field](#) by changing the field's property settings. When data is displayed as hyperlinks, users can click them to navigate to other Web pages or objects.

1. Open a datasheet or form in [PivotTable view](#).
2. Click the row, column, or detail field that contains hyperlinks.
3. On the **PivotTable** toolbar, click **Properties** , and then click the **Format** tab.
4. Select the **Display as hyperlink** check box.

[Show All](#)

Plot direct record values in PivotChart view

By default, totals from your source data are plotted in a [PivotChart view](#). The **Plot detail records** option allows you to plot values from individual records rather than the totals for groups of records.

1. Open a datasheet or form in PivotChart view.
2. On the **PivotChart** toolbar, click **Properties** , and then click the **General** tab.
3. Make sure **Chart Workspace** is selected in the **Select** list.
4. Click the **Data Details** tab.
5. Select the **Plot detail records** check box.

Caution Selecting and clearing the **Plot detail records** check box clears formatting and layout on the chart.

[Show All](#)

About sharing an Access database on a network (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

If your computer is connected to a network, you and others can work with one [Microsoft Access database](#) at the same time.

[Ways to share data](#)

There are several ways you can share data in a multiuser environment.

Share the entire Access database You can put the entire Access database on a network server or in a shared folder. This is the easiest method to implement. Everyone shares the data and uses the same forms, reports, queries, macros, and modules. Use this strategy if you want everyone to use the Access database the same way or if you can't support users creating their own objects.

Share only the tables in the Access database You can put only the tables on a network server, and keep other database objects on users' computers. In this case, the Access database's performance is faster because only data is sent across the network. When you split a database into a back end (tables) and a front end, users can customize forms, reports, and other objects in their front-end databases without affecting other users.

Share Access database objects or data access pages on the Internet You can output one or more database objects to static [HTML](#) or [server-generated HTML](#) format, or create [data access pages](#), and then display them in a browser, such as Microsoft Internet Explorer, on the [World Wide Web](#).

Replicate the Access database If you use two computers, such as an office computer and a portable computer, you can use Microsoft Windows Briefcase to make [replicas](#) of your Access database and keep those replicas [synchronized](#). Also, several users at different locations can work on their own copies at the same time and then synchronize them over the network, either through a dial-up

connection or on the Internet.

Create a client/server application If you work in a client/server environment, you can take advantage of the extra power and security it provides by creating a client/server application. You store your data in tables on a database server such as Microsoft SQL Server instead of in local tables in Microsoft Access. The Access application (the client) retrieves the data it needs from the server. The server maintains data integrity and runs any queries that it can evaluate.

[The locking information \(.ldb\) file](#)

When you open an Access database file (.mdb) in shared mode, Microsoft Access also creates a locking information file (.ldb) with the same file name (for example, Northwind.ldb) and in the same folder as the database file. This locking information file stores the computer name (such as mypc) and security name (such as Admin) of each shared user of the database. Microsoft Access uses this information to control concurrency. In most cases, Microsoft Access automatically deletes the locking information file when the last user closes the database file.

[Editing data in a shared database](#)

In a multiuser environment, more than one person might be working with the same records at the same time. Since other people can change or even delete the same data you're trying to edit, you might occasionally conflict with others as they work.

Microsoft Access helps you keep track of the status of records as you edit them, and makes sure you're using the latest data. When two or more people try to edit the same record, Microsoft Access displays messages that help you resolve conflicts. For example, if you try to save a record that another user has locked, Microsoft Access displays the name of the person who locked that record.

To help you keep track of the status of records, Microsoft Access displays the following symbols in the current [record selector](#).

Symbol

Meaning



This record is the current record and hasn't been edited.

You have edited this record, but you haven't saved your changes yet. As



long as this symbol is displayed, other users can't see the changes you've made to the record, and they won't be able to edit the record, if you have the record locked. To free the record for use by others, either save or undo your changes.



This record is locked by another user. You can't edit it. If you try to type in a locked record, Microsoft Access sounds a beep.

You can also set options that control the locking of data and the refreshing of data in a shared database.



[Default record locking settings](#)

- **No Locks** Microsoft Access does not lock the record you're editing. When you try to save changes to a record that another person has also changed, Microsoft Access displays a message giving you the options of overwriting the other user's changes to the record, copying your version of the record to the Clipboard, or discarding your changes. This strategy ensures that records can always be edited, but it can create editing conflicts between users.
- **Edited Records** Microsoft Access locks the record you're editing, so no other user can change it. It might also lock other records that are stored nearby on your disk. If another user tries to edit a record that you've locked, Microsoft Access displays the locked record indicator  in the other user's datasheet. This strategy ensures that you can always finish making changes that you start. It is a good choice if you don't have editing conflicts often.
- **All Records** Microsoft Access locks all records in the form or datasheet (and underlying tables) you're editing for the entire time you have it open, so no one else can edit or lock the records. This strategy is very restrictive, so choose it only when you know you're the only person who needs to edit records at any one time.

When you edit data in a linked [SQL database](#) table by using [ODBC](#), Microsoft Access doesn't lock records; instead, the rules of that SQL database govern locking. In this instance, regardless of the record-locking setting you choose for your database, Microsoft Access always acts as though the **No Locks** setting has been selected.



[Page-level and record-level locking](#)

You can specify the granularity of locking that Access uses in a shared database. If you use page-level locking, Access locks the 4K page (the area of memory where the record is located), and editing a record might cause other records stored nearby to be locked as well. However, performance is generally better when you use page-level locking.

If you use record-level locking, Access locks only the record you are editing. This becomes the default behavior for access to data through a [form](#), a datasheet, and code that uses a [recordset](#) object to loop through records, but not through action queries, nor through code that performs bulk operations using SQL statements.

[Update retry and refresh interval settings](#)

You can use the **Update Retry Interval** and **Number of Update Retries** settings to specify how often and how many times Microsoft Access tries to save a record that is locked by another user. You can also use the **ODBC Refresh Interval** and **Refresh Interval** settings to control how often Access refreshes your data. Refreshing only updates data that already exists in your datasheet or form. It doesn't reorder records, display new records, or remove deleted records and records that no longer meet specified criteria. To view these changes, you must requery the underlying records for the datasheet or form.

[Saving design changes in a shared database](#)

You cannot save design changes to a [Microsoft Access database](#) while other users have the Access database open. The only way to ensure that changes are saved is by opening the Access database in [exclusive](#) mode.

In general, when you try to make a design change to a [database object](#) (except tables and queries) or an item in shared mode, Access will temporarily promote you to exclusive mode for the Access database if you are the only user of the Access database at that time. When you save all your design changes and close all the [Design view](#) windows, Access returns the Access database to shared mode. In the interim, other users will not be able to open the Access database.

If another user has the Access database open in shared mode and you try to make a major design change, such as modifying a form, Access alerts you that you might not be able to save your changes. However, if another user has the Access

database open in shared mode and you try to make a minor design change, such as changing printer settings, Access doesn't alert you that you might not be able to save your design changes. In both cases, you might want to wait until you are the only user of the Access database so that you can save your major design changes and Access can save your minor design changes.

A [data access page](#) is handled differently. Although creating, renaming, moving, and deleting a data access page still requires promotion to exclusive mode (because this requires changing information in the Access database), editing a data access page does not require promotion to exclusive mode (because the corresponding HTML file exists in the file system outside the database).



[Differences between major and minor design changes](#)

If you don't have exclusive access to an Access database, Access will alert you when you might not be able to save the following major design changes:

- Changes to database objects (except table and queries) made in Design view
- Changes to a form property sheet in [Form view](#)
- Compiling the [project](#), modifying project properties, or adding or removing a reference in the Visual Basic Editor
- Renaming, pasting, or deleting a database object
- Saving a database object as another type of database object
- Adding or modifying controls on a command bar
- Editing custom groups in the [Database window](#)
- Creating, renaming, moving, and deleting a data access page

If you don't have exclusive access to an Access database, Access doesn't alert you when you might not be able to save the following minor design changes:

- Changes to datasheet formatting properties, such as line styles and fonts
- Freezing, unfreezing, hiding, and showing datasheet columns
- Adjusting datasheet column width and row size
- Changes to the filter or sort order for a form or datasheet
- Changes to the state of a subdatasheet (whether expanded or collapsed)
- Changes to the [OLE/DDE link](#) of an [OLE](#) object or changes to the contents of an [unbound object frame](#)
- Changes to printer settings

- Changes to a command bar, such as its location and whether or not it's visible
- Changes to the layout of the Database window

[Strategies for sharing Access database design and development](#)

If you are going to share the design of an Access database, consider using the following strategies:

- Assign specific objects or groups of objects to specific developers. For example, assign one developer to design forms and another to design reports. Then, each developer can work on a private copy of the Access database. When your development team is ready for testing, assembly, and production, each developer can export the database objects from their private copy of the database to a main Access database.
- Use a source code control program. If you use Visual SourceSafe, it's a good idea to run the Performance Analyzer (on the **Tools** menu, point to **Analyze**, and then click **Performance**) on a regular basis to maintain optimum performance.

[Tips for saving design changes in a shared database](#)

- If another user has a table open or is viewing data in queries, forms, or reports based on the table, you can't make any changes to the table's design. Microsoft Access informs you that it is read-only.
- If objects are dependent on one another, update all of them at the same time so that other users don't open inconsistent versions of the objects. For example, if you want to add fields to a form, make sure to add the fields to the underlying query before you update the form.
- If you change a query that another user has open, that user must close and reopen the changed object to use the latest version. With Microsoft Visual Basic for Applications (VBA) modules, other users can't run the updated **Function** or **Sub** procedures until they close and reopen the Access database.

[Show All](#)

About saving design changes to a shared Access database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You cannot save design changes to a [Microsoft Access database](#) while other users have the Access database open. The only way to ensure that changes are saved is by opening the Access database in [exclusive](#) mode.

In general, when you try to make a design change to a [database object](#) (except tables and queries) or an item in shared mode, Access will temporarily promote you to exclusive mode for the Access database if you are the only user of the Access database at that time. When you save all your design changes and close all the [Design view](#) windows, Access returns the Access database to shared mode. In the interim, other users will not be able to open the Access database. If you are in the Visual Basic Editor (VBE), you also need to save all open modules and close the VBE.

If another user has the Access database open in shared mode and you try to make a major design change, such as modifying a form, Access alerts you that you might not be able to save your changes. However, if another user has the Access database open in shared mode and you try to make a minor design change, such as changing printer settings, Access doesn't alert you that you might not be able to save your design changes. In both cases, you might want to wait until you are the only user of the Access database so that you can save your major design changes and Access can save your minor design changes.

A [data access page](#) is handled differently. Although creating, renaming, moving, and deleting a data access page still requires promotion to exclusive mode (because this requires changing information in the Access database), editing a data access page does not require promotion to exclusive mode (because the corresponding HTML file exists in the file system outside the database).



[Differences between major and minor design changes](#)

If you don't have exclusive access to an Access database, Access will alert you when you might not be able to save the following major design changes:

- Changes to database objects (except table and queries) made in Design view
- Changes to a form property sheet in [Form view](#)
- Compiling the [project](#), modifying project properties, or adding or removing a reference in the Visual Basic Editor
- Renaming, pasting, or deleting a database object
- Saving a database object as another type of database object
- Adding or modifying controls on a command bar
- Editing custom groups in the [Database window](#)
- Creating, renaming, moving, and deleting a data access page

If you don't have exclusive access to an Access database, Access doesn't alert you when you might not be able to save the following minor design changes:

- Changes to datasheet formatting properties, such as line styles and fonts
- Freezing, unfreezing, hiding, and showing datasheet columns
- Adjusting datasheet column width and row size
- Changes to the filter or sort order for a form or datasheet
- Changes to the state of a subdatasheet (whether expanded or collapsed)
- Changes to the [OLE/DDE link](#) of an [OLE](#) object or changes to the contents of an [unbound object frame](#)
- Changes to printer settings
- Changes to a command bar, such as its location and whether or not it's visible
- Changes to the layout of the Database window



[Strategies for sharing Access database design and development](#)

If you are going to share the design of an Access database, consider using the following strategies:

- Assign specific objects or groups of objects to specific developers. For example, assign one developer to design forms and another to design reports. Then, each developer can work on a private copy of the Access database. When your development team is ready for testing, assembly, and production, each developer can export the database objects from their

private copy of the database to a main Access database.

- Use a source code control program. If you use Visual SourceSafe, it's a good idea to run the Performance Analyzer (on the **Tools** menu, point to **Analyze**, and then click **Performance**) on a regular basis to maintain optimum performance.

[Tips for saving design changes in a shared database](#)

- If another user has a table open or is viewing data in queries, forms, or reports based on the table, you can't make any changes to the table's design. Microsoft Access informs you that it is read-only.
- If objects are dependent on one another, update all of them at the same time so that other users don't open inconsistent versions of the objects. For example, if you want to add fields to a form, make sure to add the fields to the underlying query before you update the form.
- If you change a query that another user has open, that user must close and reopen the changed object to use the latest version. With Microsoft Visual Basic for Applications (VBA) modules, other users can't run the updated **Function** or **Sub** procedures until they close and reopen the Access database.

[Show All](#)

Share a database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To access the shared [Microsoft Access database](#) from another computer, you must have one of following on the other computer: a local installation of Access, a network installation of Access (licensed on a per-user basis), or a run-time application.

1. Set up a shared folder.

For information about sharing folders, use the Microsoft Windows Help Index. If the shared folder is on a network server, you might need assistance from your network administrator.

2. Copy the Microsoft Access database to the shared folder.
3. Make sure the Access database is set to open in shared mode, which is the default setting.

 [How?](#)

1. On the **Tools** menu, click **Options**.
2. On the **Advanced** tab, under **Default open mode**, click **Shared**.

Note Before you share the database, you should remove any personal information that may be stored in the file. See [Remove personal information from an Access file or page](#) for more information.

[Show All](#)

Split an Access database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can split a [Microsoft Access database](#) into two files: one that contains the tables and one that contains the queries, forms, reports, macros, modules, and shortcuts to [data access pages](#). This way, users who need to access the data can customize their own forms, reports, pages, and other objects while maintaining a single source of data on the network.

1. On the **Tools** menu, point to **Database Utilities**, and then click **Database Splitter**.
2. Follow the instructions in the **Database Splitter Wizard** dialog boxes.

[Show All](#)

Refresh or requery data

If you are using a [Microsoft Access project](#), or a [Microsoft Access database](#) that's shared on a network, other users could be changing data while you are viewing the same data in a datasheet or form. Microsoft Access updates the data you see at regular intervals. However, you can immediately display the most current data by refreshing the records.

In an Access database (.mdb), refreshing records only updates the data that already exists in your datasheet or form. It doesn't reorder records, display new records, or remove deleted records and records that no longer meet specified criteria. To perform those tasks, requery the records.

Note In an Access project (.adp), the refresh and requery actions both requery the records.

To requery or refresh:

1. Open a table, query, or form in [Datasheet view](#), [PivotTable view](#), or [PivotChart view](#), or open a form in [Form view](#).
2. Do one of the following:
 - To refresh the records in Datasheet or Form view, click **Refresh** on the **Records** menu.
 - To refresh the records in PivotTable or PivotChart view, click **Refresh**  on the **PivotTable** or **PivotChart** toolbar.
 - To requery the records, press SHIFT+F9.

[Show All](#)

Set options for a shared Access database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. On the **Tools** menu, click **Options**.
2. Click the **Advanced** tab.
3. Do one or more of the following:

[Specify whether a Microsoft Access database opens in shared or exclusive mode by default](#)

- Under **Default open mode**, do one of the following:

If you want others to be able to open the [Microsoft Access database](#) at the same time you have it open, click **Shared**.

If you want sole access to the Access database when you have it open, click **Exclusive**.

Notes

- Even if you have set the default open mode to **Shared**, you can temporarily override this setting at the time you open an Access database.
- If you set up security in your multiuser environment, you can help protect from users opening an Access database exclusively by denying them Open Exclusive [permission](#) for the database.

[Specify update retry and refresh interval settings](#)

- Do one or more of the following:

To set the interval after which Microsoft Access automatically tries to save a changed record that is locked by another user, type the number

of milliseconds in the **Update retry interval** box. Valid values are 0 through 1,000.

To set the number of times Microsoft Access tries to save a changed record that is locked by another user, type a number in the **Number of update retries** box. Valid values are 0 through 10.

To set the interval after which Microsoft Access automatically refreshes records that you're accessing using [ODBC](#), type the number of seconds in the **ODBC refresh interval** box. Valid values are 1 through 32,766.

To specify how often data is automatically updated, type the number of seconds in the **Refresh interval** box. Valid values are 1 through 32,766 seconds. For example, if you type 120, the data is updated every 2 minutes. This setting takes effect only if the database is shared on a network.

[Specify the record locking strategy](#)

When a user edits a record, Microsoft Access can automatically prevent others from changing that record until the user has finished editing it. Giving one user exclusive access to a record is called *locking*.

- Under **Default record locking**, do one of the following:

To prevent locking of records while you edit them, click **No locks**.

To lock all records in the form or datasheet (and the underlying tables) while you edit them, click **All records**.

To lock only the record you're editing, click **Edited record**.

[Specify record-level or page-level locking](#)

- Do one of the following:

To make record-level locking the new default setting for the current [Microsoft Access database](#), select the **Open databases using record-level locking** check box.

To make page-level locking the new default setting for the current Access database, clear the **Open databases using record-level locking** check box.

This setting takes effect the next time you open the Access database, but you must use the **Open** command on the **File** menu rather than the list of most recently used files at the end of the **File** menu.

[Show All](#)

Troubleshoot a shared database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[I can't save changes to a Microsoft Access database.](#)

You cannot save design changes to a shared [Access database](#) while other users have the database open. The only way to ensure that changes are saved is by opening the Access database in [exclusive](#) mode.

[I can't change the table's design.](#)

- If another user has a table open or is viewing data in queries, forms, or reports based on the table, you can't make any changes to the table's design. Microsoft Access informs you that it is read-only.
- You might not have [permission](#) to change the design. If the shared [Microsoft Access database](#) you're using employs security measures, access to objects (except [modules](#)) is controlled by the people who create or administer it. Your permissions might be determined by the [group](#) you are in.

[I can't view data in a table.](#)

- You can't view data in a table that another user has open in [Design view](#). Wait until the table is available. When you open a table in Design view, keep it open only as long as necessary if others need to use the table or any queries, forms, or reports based on the table.
- You might not have [permission](#) to view the data. If the shared [Microsoft Access database](#) you're using employs security measures, access to objects (except [modules](#)) is controlled by the people who create or administer it. Your permissions might be determined by the [group](#) you are in.

[I can't open a database object.](#)

If the shared [Microsoft Access database](#) you're using employs security measures, you might not have [permission](#) to open the object. Also, you might have permission to view an object but not to see it in [Design view](#) or to update it. Your permissions might be determined by the [group](#) you are in. If necessary, contact the creator or administrator of the Access database to determine whether you have permission to open an object.

[I can't open a shared Access database.](#)

Another user is making design changes to the database. Once another user begins making design changes, the database is open in [exclusive mode](#) instead of shared mode. Even after the other user has switched from [Design view](#) to [Form](#) or [Datasheet view](#), exclusive mode remains in effect until the other user closes the object where he or she made the design changes.

[Show All](#)

Automate importing, exporting, or linking data

Depending on the type of data you're working with, you can automate the [importing](#), [exporting](#), and linking of data by carrying out the TransferDatabase, TransferSpreadsheet, or TransferText actions in a [macro](#) or Microsoft Visual Basic [procedure](#).

[Use a macro](#)

1. Create a macro.

[How?](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the **Macro Design** toolbar.
 2. In the **Action** column, click the arrow to display the [action list](#).
 3. Click the action you want to use.
 4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
 5. Type a comment for the action. Comments are optional.
4. To add more actions to the macro, move to another [action row](#) and repeat step 3. Microsoft Access carries out the actions in the order you

- list them.
2. Do one of the following to import, export, or link data between the current Microsoft Access database and:
 - **Another database** Click **TransferDatabase** in the [action list](#) of a blank action row.
 - **A spreadsheet file** Click **TransferSpreadsheet** in the action list of a blank action row.
 - **A text file** Click **TransferText** in the action list of a blank action row.
 3. Set the [action arguments](#) as appropriate for the action.
 4. Click **Save** to save the macro.
 5. Run the macro.

[How?](#)

To run a macro directly, do one of the following:

- To run a macro from the [Macro window](#), click **Run**  on the **Macro Design toolbar**.
- To run a macro from the [Database window](#), click **Macros**  and then double-click a macro name.
- On the **Tools** menu, point to **Macro**, click **Run Macro**, and then select the macro from the **Macro Name** list. Then click a macro in the **Macro Name** box.
- Run a macro from a [Microsoft Visual Basic](#) procedure by using the **RunMacro** method of the **DoCmd** object.

[Use a Visual Basic procedure](#)

1. Create a Visual Basic procedure.

[How?](#)

1. To open a [module](#), do one of the following:
 - To open a new [standard module](#), in the Database window, click **Modules**  under **Objects**, and then click the **New** button on the

Database window toolbar.

- To open an existing standard module, click **Modules**  under **Objects**, select the module you want to open, and then click **Design**.
 - To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the **Form Design** toolbar.
 - To open a new [class module](#) that isn't associated with a form or report, in the Database window click **Class Module** on the **Insert** menu.
 - To open an existing class module, in the Database window, click **Modules**  under **Objects**, select the module you want to open, and then click the **Design** button on the Database window toolbar.
2. Declare the function by typing the **Function** statement.
 3. Type a function name immediately followed by any function [arguments](#) in parentheses. For example, the following declaration for the **IsLoaded** function specifies *strFormName* as an argument:

```
Function IsLoaded (strFormName As String) As Boolean
```

4. Add the [Microsoft Visual Basic](#) code that performs the operation or calculation that you want the function to perform.
2. Do one of the following to import, export, or link data between the current Microsoft Access database and:
 - **Another database** Use the **TransferDatabase** method to carry out the TransferDatabase action in the procedure.

[How?](#)

Microsoft Access defines a special object, **DoCmd**, that you use to carry out macro actions in [Microsoft Visual Basic](#) procedures. You carry out an [action](#) by adding a [method](#) of the **DoCmd** object to your procedure. Most actions have a corresponding **DoCmd** method.

- In the procedure, add the **DoCmd** method that corresponds to the action by using the following syntax:

```
DoCmd.method [arguments]
```

Method is the name of a method. *Arguments* are the method arguments, if there are any.

For example, to create a procedure that carries out the OpenForm action, add the **OpenForm** method of the **DoCmd** object to the procedure. The following method is equivalent to the OpenForm action; it opens the Add Products form:

```
DoCmd.OpenForm "Add Products"
```

A few actions don't have corresponding **DoCmd** methods, although some have equivalent Visual Basic statements or functions.

 [Macro actions that don't have a corresponding DoCmd event](#)

Action	Microsoft Visual Basic equivalent
--------	-----------------------------------

AddMenu	No equivalent
MsgBox	MsgBox function
RunApp	Shell function
RunCode	Procedure call (Call statement)
SendKeys	SendKeys statement
SetValue	Assignment statement (Let statement)
StopAllMacros	Stop or End statements
StopMacro	Exit Sub or Exit Function statements

- **A SQL Server database** Use the **TransferSQLDatabase** method to transfer an entire Microsoft SQL Server database to another SQL Server database.
 - **A spreadsheet file** Use the **TransferSpreadsheet** method to carry out the TransferSpreadsheet action in the procedure.
 - **A text file** Use the **TransferText** method to carry out the TransferText action in the procedure.
 - **An XML file** Use the **ExportXML** or **ImportXML** method to export to or import from an XML file, respectively.
3. Click the **Run Sub/UserForm** command on the **Run** menu in the Visual Basic Editor to run the procedure.

[Show All](#)

About XML data and Access

Some of the content in this topic may not be applicable to some languages.

[Extensible Markup Language](#) (XML) is the standard language for describing and delivering data on the [Web](#), just as [Hypertext Markup Language](#) (HTML) is the standard language for creating and displaying Web pages. Microsoft Access provides ways to both import and export XML data as well as transform the data to and from other formats using XML related files.

 [About XML](#)

Moving information across the Internet and between software applications has always been difficult due to differences in data formats and proprietary structures. HTML, while well suited for providing text and image display information for Web browsers, is limited in its ability to define data and data structures. HTML describes how a Web page should look, whereas XML defines the data and describes how the data should be structured. XML is a data interchange format in that it allows you to exchange data between dissimilar systems or applications. XML separates the data from the presentation so that the same XML data can be presented in multiple ways by using different presentation files.

XML is a standards-based protocol that is governed by the [World Wide Web Consortium](#) (W3C). The XML protocol is a set of rules, guidelines, and conventions for designing data formats and structures, in a way that produces files that are easy to generate and easily read by different computers and applications. In addition, XML defined structures are unambiguous; that is, they are self-describing and platform-independent. Like HTML, XML makes use of tags and attributes, but while HTML specifies what each tag and attribute means (and thus how the data between them will look in a browser), XML uses the tags only to delimit pieces of data, and leaves the interpretation of the data completely to the application that reads it. Additional information about XML can be found on the [MSDN](#) Web site.

 [About importing XML data](#)

Access provides choices for using data from many external sources. XML makes it easier to transform the data from almost any external application for use by Access. You can:

- Import XML data to a [Jet](#), [Microsoft SQL Server](#), or Microsoft SQL Server 2000 Desktop Engine (formerly called MSDE) database.
- Import XML schema data to a Jet, SQL Server, or Microsoft SQL Server 2000 Desktop Engine database.
- Use an Extensible Stylesheet Language Transformation (XSLT) file to transform the data into an XML format that Access supports.

Note Importing XML into SQL Server or Microsoft SQL Server 2000 Desktop Engine database through an Access project (ADP) requires that Microsoft SQL Server 2000 Desktop Engine is installed locally.

You can use the **Import** command (point to **Get External Data** on the **File** menu) to import XML data files into Access. This command displays the **Import** dialog box so that you can select an XML document as well as a schema, which describes the structure of the data. You can only import a single document at a time into Access. The data must be in a format that Access recognizes, either in a native format or through the use of a schema. Note that when importing XML data, you cannot choose a subset of the XML document; the entire file has to be imported.

You can transform any XML data into a format that Access supports by selecting a transform in the **Import XML** dialog box. You can also specify whether to overwrite any existing tables or append to existing data.



[What are XML schemas?](#)

You use XML schemas to describe the structure of data in a common format that customers, other Web browsers, and any number of XML-enabled software programs can recognize. Specifically, schemas define the rules of an XML data document, including element names and data types, which elements can appear in combination, and which attributes are available for each element. Schemas provide a model for an XML data document which defines the arrangement of tags and text within all documents referencing the schema. Access supports the XML Schema standard (XSD). XSD is an approved [World Wide Web Consortium](#) (W3C) standard designed as a basic infrastructure for describing the

type and structure of XML documents.

By using a schema, you can ensure that any XML document that is used to import data into Access or export from Access to another format contains specific data and conforms to a defined structure. You can also provide the schema to other businesses and applications so that they know how they should structure any data they provide to you and they, in turn, can provide their schema to you.

[Displaying XML data](#)

XML syntax (the tags and their placement in a document) defines and describes the data in an XML document but doesn't indicate how the data should be displayed. In earlier applications and in some current uses, Cascading Style Sheets (CSS) were used to tell the browser how to display the XML data. However, a CSS isn't always a good choice because they are written in a specialized language which means that the developer has to use another language in order to write, modify, or understand the structure of the style sheet. A CSS file merely allows you to specify the formatting of each XML element without much control over the output.

On the other hand, an Extensible Stylesheet Language (XSL) style sheet is considerably more flexible than a CSS file. It allows you to precisely select the data that will be displayed, to specify the order or arrangement of the data, and to modify or add information. Additionally, it is written in a style similar to an XML document using a combination of XML-like tags and HTML to create a template for a specific style of output. Additional information about XSL style sheets can be found on the [MSDN](#) Web site.

Note that you are not required to link either a CSS file or an XSL style sheet to an XML document in order for Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later to display the document. Internet Explorer has a default, built-in style sheet that displays the XML source as a collapsible/expandable tree.

You can use style sheets to insure that the XML-based Web pages on your intranet or Website are consistent and present a uniform appearance without having to add HTML to each page.

[About Extensible Stylesheet Language Transformation \(XSLT\)](#)

XSL for Transformation (XSLT) is a specification that is recommended by the [World Wide Web Consortium](#) (W3C), and supported by Access. XSLT is an XML-based language that allows one XML document to be mapped, or transformed, into another XML document. This provides a way of transforming an XML document's presentation information from a source format to a target format and back again. Typically, a developer creates an XSL transformation file that, when applied to an XML document during export, interprets or transforms the XML data into a presentation format that can be recognized by another application, such as Service Advertising Protocol (SAP) or a custom purchase order format.

XSLT has many of the constructs (structures and commands) found in other programming languages which allow the developer to use variables, loops and iterations, and conditional statements. This gives the developer considerable control over the output of the XML data. Additional information about XSLT can be found on the [MSDN](#) Web site.

[About exporting to XML files](#)

Exporting data and database objects to an XML file is a convenient way to move and store your information in a format that can readily be used across the Web. In Access, you can export the data, the schema (data structure), or both, to XML files. You can:

- Export data to an XML file and, optionally, use an XSLT to transform the data to another format.
- Export the data schema using XML Schema standard (XSD).
- Export the data behind forms and reports to an XML file.

You can also transform the data to another presentation format using an Extensible Style Language (XSL) file during the export process. You can export tables, queries, and the data behind forms or reports from a [Microsoft Access database](#) (.mdb) as well as tables, queries, stored procedures, functions, and the data behind forms and reports from a [Microsoft Access project](#) (.adp).

[Exporting tables, queries, views, datasheets, forms or reports](#)

You can export a database object as an XML document in several ways:

- You can export just the data from a table, query, datasheet, form, or report into an XML file. This data is saved to a file named **<filename>.xml**.

Note When you export a table to an XML document, you can also export related tables. For example, if you export a table of Customers Orders, you can also choose to export a related Orders Details table and Customers table into the same file.

- You can export just the schema (data structure) of a table, query, datasheet, form, or report to an XML schema file. An XML schema file is a formal specification of the rules for an XML document, providing a series of element names, as well as which elements are allowed in the document and in what combinations. If you select to save the schema as XSD, the file is saved as **<filename>.xsd**.
- In addition, when you choose to save the data as XML, you can specify that the data be transformed to a custom display format by using an existing .xsl file. If no .xsl file is specified, the data is saved in standard XML format. Note that if no data is selected for export then a presentation format is also unavailable. The file is saved as **<filename>.xsl**.
- Through the Access object model, you can also save the presentation attributes of a table, query, datasheet, form, or report into a file that describes the presentation and connection information. For forms and reports, this file is saved in an XML-based language called ReportML which provides presentation data as well as a data model for creating a [data access page](#). For tables, queries/views, and datasheets, the presentation file is a spreadsheet-like template. This file is saved as **<filename>_report.xml**.

[What is ReportML?](#)

ReportML is a "language" developed by Microsoft and specific to Access which can be used to describe Access database objects in XML. The ReportML language is made up of a set of tags that describe a form, report or data access page's properties, events, and attributes. When you export data from Access to an XML file, you choose to save the structure of a form or report into a ReportML format. The ReportML file can be used to convert the saved data into a data access page.

[Well-formed documents](#)

Any XML document produced by Access is well-formed, which means that it conforms to the basic rules of XML. That is:

- Each XML document must have a unique root element (an element encompassing the entire document).
- All start and end tags match. XML tags are case-sensitive.
- For each start tag, there is a corresponding end tag. Empty elements can be denoted by a special shorthand tag. Again, XML tags are case-sensitive.

Note Access ReportML will not write out any empty tags.

- Elements do not overlap. In other words, start and end tag must be properly nested within other elements.
- Certain reserve characters are part of the XML syntax and will not be interpreted as themselves if used in the data portion of an element. You need to substitute a special character sequence (called an "entity" by XML) as follows:

Character Entity

<	<
&	&
>	>
"	"
'	'

The following is an example of a well-formed XML document:

```
<?xml version="1.0"?>
<ORDER>
  <CUSTOMER>Nancy Davolio</CUSTOMER>
  <PRODUCT>
    <ITEM>Mom&apos;s Boston Crab Meat</ITEM>
    <PRICE>$10.00</PRICE>
    <QUANTITY>1 Bottle</QUANTITY>
  </PRODUCT>
</ORDER>
```

where:

<?xml version="1.0"?> is a declaration that states that this is an XML document and gives the version number. The declaration optional but is recommended in any XML document.

<ORDER> is the root element which encompasses the entire document.

<CUSTOMER> is a start tag and </CUSTOMER> is an end tag which together describe an element of data, in this case, the customer's name.

Notice that each tag set, such as <CUSTOMER></CUSTOMER>, has both start and end tags and is case sensitive, and that the tag sets are properly nested within each other. Also notice the entity ' which will be transformed to an apostrophe (') when the data is imported by the receiving application. The apostrophe has a special purpose in an XML document and can be misinterpreted if used directly in the text. The converted data will be displayed as *Mom's Boston Crab Meat*.

White space can be used throughout the document to enhance readability.

If a schema is specified while exporting from Access, then the XML documents created are considered to be valid XML document. This means that in addition to being well-formed, the documents conform to a defined schema.

[Show All](#)

About exporting data and database objects

Exporting is a way to output data and database objects to another database, spreadsheet, or file format so another database, application, or program can use the data or database objects. Exporting is similar in functionality to copying and pasting. You can [export](#) data to a variety of supported databases, programs, and file formats. You can export most [database objects](#) from a [Microsoft Access database](#) or [Microsoft Access project](#) to another Access database or Access project.

 [Exporting database objects.](#)

You cannot export [database diagrams](#) to an Access project or an Access database. You cannot export [queries](#) or relationships from an Access database to an Access project.

When exporting data to an XML document, you can export the main table and any related tables. When exporting to another database, you can only export one object at a time. If you want to export multiple objects to a database, open the database you want to export to, and then [import](#) the objects instead.

When you export an object to another database, you might also want to export related objects to make it work. For example, a report might have a query as a [record source](#), and that query might be based on a relationship between two tables.

When you export a [data access page](#), you are exporting only the link to the corresponding [HTML](#) file. The HTML file remains in the same location, which means you have two data access pages pointing to the same corresponding HTML file. If you don't want this, save one of the data access pages under a different name and enter a different file location for the new corresponding HTML file.

You can use the Transfer Database wizard to create a new SQL Server database and export the objects from an existing Access database. See Help for more

information on the Transfer Database wizard.

[Mailing database objects.](#)

You can e-mail the output of a database object by attaching it to an electronic mail message in several different file formats, including Microsoft Excel (.xls), Rich Text Format (.rtf), MS-DOS text (.txt), [HTML](#) (.html), [IDC/HTX](#), [ASP](#), and [data access pages](#) (.html). You can also send a page as the body of an e-mail message.

To send a database object in e-mail, you need Microsoft Outlook, Microsoft Exchange, Microsoft Mail, or another electronic mail application that supports Messaging Application Programming Interface (MAPI).

[Considerations when sending a data access page in e-mail.](#)

When you send a [data access page](#) in e-mail, there are important connection and security considerations you need to be aware of to ensure more reliable results.

[Share the database.](#)

A data access page is connected to one of two types of data sources: a [Microsoft Access database](#) or a [Microsoft SQL Server database](#). For the page to work properly when sent via e-mail, the source database must be on a shared server or shared computer in order for users to view and interact with the page.

[Use UNC paths.](#)

Before you create the page, open the [Microsoft Access database](#) by using a [universal naming convention \(UNC\)](#) path in the **File name** box on the **Open** dialog box (**File** menu, **Open** command). This ensures that the data source of the page is defined with a UNC path instead of with the drive letter of a mapped network drive in Microsoft Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for the page to be located.

When you create the page, you should save it using a UNC path in the **File name** box of the **Save As Data Access Page** dialog box. Saving your page with a UNC path helps protect you when you send a pointer to the page by e-mail.

[Store the database and the page on the same server.](#)

To further help protect your pages, put the [Microsoft Access database](#) or [Microsoft SQL Server database](#) on the same [Web server](#) as the data access page.

Note You can't copy an Access database to or create an Access database in a folder under Network Places.

[Publish from a trusted intranet security zone.](#)

Depending on the Microsoft Internet Explorer security level setting on the recipient's computer, one of three things occurs when the recipient opens an e-mail message containing a data access page: the page is automatically disabled, the recipient is asked whether to allow data access, or the page is automatically enabled.

Note It is recommended that you set your Internet Explorer security setting for **Access data sources across domains to Prompt**.

If you are using a corporate intranet, you can publish your pages from a Web site located in a trusted security zone. This will help alleviate problems caused by the recipient's Internet Explorer security settings and thus improve performance of your pages. For more information about security levels and security zones, see Internet Explorer Help. For any specific page security requirements at your site, see your system administrator.

[Send a pointer instead of a copy of the HTML source code.](#)

There are three ways Microsoft Access can send a data access page in an e-mail message.

- In the body of the message, as a copy of the HTML source code of the page
- In the body of the message, as a pointer to the HTML file associated with the page
- As an attachment to the message, including a copy of the HTML file associated with the page

In the e-mail message, a copy of the HTML source code of a page looks very different than a pointer to the page. The copy will contain all of the HTML tags

and objects needed to render the page. The pointer will contain some HTML tags and the absolute path— a [universal resource locator \(URL\)](#), or a [universal naming convention \(UNC\)](#) path— to the page.

It's best to avoid sending a copy of the HTML source code of a page because Microsoft Outlook won't let a user connect to data across domains unless the recipient or system administrator has changed the default Internet Explorer security zone settings. Sending a pointer to the page assumes that the page and the data source that the page connects to are located on a server or public share. This reduces the chance for cross-domain issues within Microsoft Outlook.

Sending a pointer instead of a copy of the page can also be a security precaution that reduces the risk of malicious users intercepting a copy of the HTML source code of a page that has been sent to a user, and then revising it to covertly gain access to a corporate database.

The best way to send a data access page is to send a pointer to the page in the body of the e-mail message. To send a pointer, you can do one of the following:

- Save the page to a folder under Network Places or on a Web server, so the page's location is a URL.
- Save the page to a shared computer using a UNC path.

In either case, Access uses the <IFRAME> HTML tag to send a pointer to the page. The pointer is either the URL or the UNC path to the page.

When you create your e-mail message, Access provides confirmation dialog boxes to help you make the right choice.

To help maintain security when sending a page, first send a pointer to the page to yourself and then open the e-mail message. In the message window, right-click the page, and then click **View Source**. If the HTML text looks similar to the following code, then you are sending a pointer to the page (and not a copy of the HTML source code):

```
<HTML><BODY STYLE="OVERFLOW:HIDDEN" BOTTOMMARGIN=0  
LEFTMARGIN=0 TOPMARGIN=0 RIGHTMARGIN=0>  
<IFRAME FRAMEBORDER=0 HEIGHT=100% WIDTH=100%  
SRC="http://NWWeb/Northwind pages/View Products.htm"></IFRAME>  
</BODY><HTML>
```

[Exporting to a Web page.](#)

There are four ways you can use Microsoft Access data or database objects to create a Web page:

- In a Web browser that supports [Dynamic HTML](#) version 4.0 or later, you can use [data access pages](#). You do not output data access pages; you create a data access page as a database object that is actually a link to the data access page's corresponding HTML file. Use data access pages to view, edit, update, delete, filter, group, and sort live data from either a Microsoft Access database or a Microsoft SQL Server database.
- You can output [server-generated HTML](#) files, either [ASP](#) or [IDC/HTX](#), from tables, queries, and forms. Server-generated HTML files are displayed in a table format in a Web browser. Use server-generated HTML files when you want to use any Web browser, your data changes frequently, or you need to see live data in a table connected to an [ODBC data source](#), but you don't need to update or interact with the data.
- You can output static HTML files from tables, queries, forms, and reports. In a Web browser, reports display in a report format, and tables, queries, and forms display in a datasheet format. Use static HTML files when you want to use any Web browser that supports HTML version 3.2 or later and your data does not change frequently.
- You can save a form or report as a data access page by using the **Save As** option under the **File** menu. This allows you to use the form or report as Web page.

[Exporting to or exchanging data with Microsoft Word or another word-processing program.](#)

There are three ways to use your Microsoft Access data with word-processing programs:

- In an Access database, you can use the Microsoft Word Mail Merge Wizard to create a mail merge document in Word that links to Access data. Once the link is established, you can open your document in Word at any time to print a new batch of form letters or labels using the current data in Access.
- For any version of Microsoft Word, you can export Microsoft Access data to a mail merge data source file that can be used with the mail merge feature of Word.

- You can save the output of a datasheet, form, or report as a Rich Text Format (.rtf) file or a text (.txt) file. A Rich Text Format file preserves formatting such as fonts and styles, and can be opened with Microsoft Word and other Microsoft Windows word-processing or desktop-publishing programs.

[Exporting to Microsoft Excel or another spreadsheet program.](#)

There are three ways you can use Microsoft Access data with Microsoft Excel or another spreadsheet program:

- You can export a datasheet as unformatted data to Microsoft Excel or another spreadsheet program.
- You can save the output of a datasheet, form, or report directly as an Excel (.xls) file or worksheet. This way, you can preserve most formatting, such as fonts and colors. Report [group levels](#) are saved as Excel outline levels, and a form is saved as a table of data.
- You can also export the data from a table, query/view, datasheet, form, or report to an XML file which can then be imported into Excel.

[Exporting or copying a PivotTable or PivotChart view.](#)

If you are familiar with Microsoft Excel PivotTable reports and want to work with the data in Excel, there are two ways to accomplish this.

[Exporting data to an interactive Excel PivotTable report.](#)

To view and modify the contents of a [PivotTable view](#) by using Excel, you can export the PivotTable view to Excel.

Preparing for export to Excel Excel PivotTable reports cannot display [detail fields](#). When you export to Excel, detail fields will be available on the **PivotTable** toolbar in Excel, but the fields won't be displayed in the report. If you want the Excel PivotTable report to reflect the appearance of the PivotTable view, before you export to an Excel PivotTable report, either move all the fields out of the detail area, or hide detail data for items and cells so that the detail area is not displayed.

After you export You might notice some differences in your PivotTable view

after you export it to Excel. This is because PivotTable views use the Microsoft Office PivotTable Component, and Excel PivotTable reports either do not support certain PivotTable list features, or they implement some features differently. For information about differences between PivotTable lists and Excel PivotTable reports, see Excel Help.

 [Copying data to another program for noninteractive use.](#)

Copying to Microsoft Word or Excel When you want to work with a snapshot of the data currently displayed in a [PivotTable view](#), you can copy and paste the data into Excel or Word. When you copy the whole PivotTable view to another program, or copy selected parts of the data, the text and values from the PivotTable view become part of your Word document or Excel workbook. No connection is created between the original source data and the copy in the document or workbook, so you won't be able to refresh or update the data that was copied.

Copying to other programs You can also copy data displayed in a PivotTable view to other programs, such as Microsoft FrontPage. When you copy data to a program other than Excel or Word that does not support the [HTML](#) format used by PivotTable views, the data is copied as unformatted text.

 [Exporting to Microsoft Windows SharePoint Services.](#)

You can now export the contents of a table or a query to a list in Windows SharePoint Services. After exporting the table or query to a list, you can further modify the list in Windows SharePoint Services without affecting the table or query in Access.

 [Exporting to XML.](#)

When you export data to XML, you can export just the data, the schema, or both. You can also choose to export a presentation XSL file. The file is generated based on presentation information in the Access form, report, or datasheet.

You can export static data in tables and queries, and forms and reports that have tables or queries as their data sources to XML. You can specify a transformation file and apply the transform following the export operation. You can export related tables and queries, and carry over any filter or sort order that is specified

on the exported object to the output.

You can also export live data in tables, queries, stored procedures, functions, forms and reports in Microsoft Access Projects (ADPs) to XML. These database objects have SQL server as a live data source. You can specify the names of the SQL server and the database as the virtual directory, and export these database objects to an XML file.

[Automating export operations.](#)

To make frequent export operations more convenient, you can automate them by creating a macro or creating a [Microsoft Visual Basic for Applications \(VBA\)](#) procedure. This is useful, for example, when you export data on a regular schedule or you are building a Web site by using multiple database objects.

[About import/export specifications and schema.ini files.](#)

An import/export specification contains information, such as file format, date order, or number formats, that Microsoft Access uses to import or export a fixed-width or delimited text file. An import/export specification is stored with the default name: *Filename_ImportSpec* or *Filename_ExportSpec* in the database that you import to or export from.

You create an import/export specification by using either the Import Text Wizard or the Export Text Wizard. Use an import/export specification when you want to repeatedly import to the same table, export to the same file, or automate the import or export process.

Alternatively, you can use a Schema.ini file in a Microsoft Visual Basic program to provide even more control over data in the text file, such as specifying special currency formats or handling floating point data types. A schema.ini file is a text file containing entries that override default text driver settings in the Microsoft Windows registry. You store a schema.ini in the same folder as the imported or exported text file, and it must always be named schema.ini.

[Show All](#)

Export data or database objects

To an Access or Excel file

 [Export database objects to an existing Microsoft Access database or Microsoft Access project](#)

1. In the [Database window](#), click the name of the object you want to [export](#), and then on the **File** menu, click **Export**.
2. Click the arrow to the right of the **Save in** box, and select the drive or folder where the database you want to export to is located.
3. Double-click the icon for the database that you want to export to.
4. In the **Export** dialog box, enter a name for the new object (or accept the current name). If you are exporting a table, in **Export** dialog box, select whether you want to export both the table's definition and data or just the table's definition.

 [To a Microsoft Excel or other spreadsheet](#)

Note In addition to the following methods, you can also export Access data to an [Extensible Markup Language](#) (XML) file which can then be imported into Microsoft Excel.

 [Save an object's output as a Microsoft Excel file](#)

1. In the [Database window](#), click the name of the object you want to save. To save part of a datasheet, open the datasheet and select a portion of the datasheet before continuing.
2. On the **File** menu, click **Export**.
3. In the **Save as type** box, click **Microsoft Excel 5-7** or **Microsoft Excel 97-2003**.
4. Click the arrow to the right of the **Save in** box, and select the drive or folder to save to.

5. In the **File name** box, enter a name for the file (or use the suggested name).
6. Select the **Save formatted** check box.
7. Do one of the following:
 - If you are saving a datasheet, click **Export All** to save the entire datasheet or **Save Selection** if you selected a portion of the datasheet in step 1.
 - For all other database objects, click **Export**.

[Load the output of a table, query, form, or report into Microsoft Excel](#)

1. In the [Database window](#), click the name of the datasheet, form, or report you want to save and load into Microsoft Excel. To save a selection of a datasheet, open the datasheet, and then select the portion of the datasheet before continuing.
2. On the **Tools** menu, point to **Office Links**, and then click **Analyze It with Microsoft Excel**.

[Export or copy a PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one of the following:

[Export a PivotTable view to Excel](#)

Use this procedure to export a PivotTable view of a datasheet or form to Microsoft Excel as an interactive PivotTable report. The link between the Excel PivotTable report and the source data is maintained.

For best results, either move all fields out of the [detail area](#) or hide detail data for items before you export the PivotTable view to Excel.

- On the **PivotTable** toolbar, click **Export to Microsoft Excel**  to run or switch to Excel and display the data in a PivotTable report.

Note Because PivotTable reports in Excel cannot display some PivotTable view layouts and calculations, the PivotTable report might look different from the PivotTable view. For information about working with Excel PivotTable reports and adding fields to them, see Excel Help.

[Copy a PivotTable view to Excel or Word](#)

Use this procedure to copy a PivotTable view, or a part of it, as a noninteractive, formatted table in Microsoft Excel or Microsoft Word.

1. Do one of the following:
 - To copy the entire PivotTable view, press CTRL+A.
 - To copy a part of a PivotTable view, select the elements that you want to copy.
2. On the **PivotTable** toolbar, click **Copy** .
3. Switch to Excel or Word, and click where you want the data from the PivotTable view to appear.
4. Click **Paste** .

Note When you copy a PivotTable view, or a part of it, you are copying only the formatting and data values. No link is maintained between the copy in Excel or Word and the source data. The copy in Word or Excel is noninteractive and cannot be refreshed.

[Export a spreadsheet on a data access page to a Microsoft Excel workbook](#)

1. In the design window, make sure the spreadsheet is activated by double-clicking on the spreadsheet. This will open the application that created the object, which in this case is Microsoft Excel. For additional information on activating a spreadsheet, see Excel Help.
2. On the toolbar in the spreadsheet, click **Export to Microsoft Excel**  to run or switch to Excel and display the data in a workbook.

Notes

- Because Excel only supports 256 columns and 65,536 rows, data in columns IW through ZZZ and rows 65,537 through 262,144 are not exported from the spreadsheet. You can, however, copy this data to more than one Excel worksheet.
- When you export data from a spreadsheet to Excel, only the values and formulas are exported. Links that connect to other data on the Web page or to other Web pages are not retained in Excel.
- When you export some spreadsheet number formats, the data in Excel is displayed with a custom number format. For example, the Yes/no, True/false, and On/off spreadsheet formats are displayed with custom number formats in Excel. For information about custom number formats in Excel, see Excel Help.

To a Web page

[Export to HTML format](#)

Depending on whether you need static or dynamic data, you can export to different types of [HTML](#) data in Microsoft Access.

[Export a datasheet to dynamic HTML format](#)

1. In the [Database window](#), click the name of the table, query, or form you want to export, and then on the **File** menu, click **Export**.
2. In the **Save As Type** box, click **Microsoft IIS 1-2 (*.htx; *.idc)** or **Microsoft Active Server Pages (*.asp)**.
3. Click the arrow to the right of the **Save In** box and select the drive or folder to export to.
4. In the **File Name** box, enter the file name.
5. Click **Export**.
6. Enter the appropriate information in the **HTX/IDC Output Options** or **Microsoft Active Server Pages Output Options** dialog boxes:
 - In the **HTML Template** box, enter the location of an [HTML](#) template.
 - In the **Data Source Name** box, enter the name of the [ODBC data source](#) that you will connect to when the server-generated HTML files are processed on the Web server.

7. Do one of the following:

- For a [Microsoft Access database](#) (.mdb)

In the **User to Connect As** and **Password for User** boxes, enter a user-level security username and password to allow users to access the database from the Web page. If you do not enter a username or password, the default username is **Admin**, and no password is used.

Warning This will allow anyone with default Administrator privileges to access your database from the Web page. It is highly recommended that Administrator accounts have strong passwords assigned to them. Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

- For a [Microsoft Access project](#) (.adp)

In the **Username** and **Password** boxes, enter a database username and password to allow users to access the Microsoft SQL Server database from the Web page. If you do not enter a username or password, the default username is **Sa** and no password is used.

Warning This will allow anyone with default Administrator privileges to access your database from the Web page. It is highly recommended that Administrator accounts have strong passwords assigned to them. Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

Warning The password you provide is displayed in the dialog box and stored in the file as plain text. An unauthorized user may be able to access the information in the file and compromise the security of the data source. As an alternative, you can use ODBC Data Source Administrator to add a system data source name (DSN) to your computer and configure that DSN with a user name and a password. You can then specify the system DSN as

the **Data Source Name** in the **Microsoft Active Server Pages Output Options** dialog box and leave the **Username** and **Password** fields blank. For more information about ODBC Data Source Administrator, see Microsoft Windows Help.

8. Once you have exported a file to [ASP](#) or [IDC/HTX](#) format, you need to publish the files to make them available on the [World Wide Web](#).

[Export a datasheet to static HTML format](#)

1. In the [Database window](#), click the name of the table, query, or form, you want to export, and then on the **File** menu, click **Export**.
2. In the **Save As Type** box, click **HTML Documents (*.html;*.htm)**.
3. Click the arrow to the right of the **Save In** box and select the drive or folder to export to.
4. In the **File Name** box, enter the file name.
5. Select the **Save Formatted** check box to:
 - save the table, query, or form in a format similar to its appearance in [Datasheet view](#),
 - enable the **AutoStart** check box,
 - and display the **HTML Output Options** dialog box after you click **Save** in step 7.
6. Select **AutoStart** if you want to display the results in your default [Web browser](#).
7. Click **Save**.
8. If you selected **Save Formatted** in step 5, the **HTML Output Options** dialog box is displayed. In the **HTML Template** box, you can enter the location of an [HTML](#) template. You can also set encoding options (**Default encoding**, **Unicode**, and **Unicode (UTF-8)**) for saving the file.
9. Once you have output a datasheet to an HTML file, you need to [publish](#) it to make it available on the [World Wide Web](#).

Notes

- If the datasheet contains a [parameter query](#), Microsoft Access first prompts you for the parameter values, then exports the results.
- Fields with a Hyperlink data type are exported as hyperlink addresses by using <A HREF> HTML [tags](#).

- Although a default [theme](#) is not applied when you export the datasheet, you can apply a theme by creating a data access page from an existing Web page (in this case, the static HTML file you just created), applying a theme in page Design view, and then saving the page.
- A datasheet with many records may take a long time to export and display through a Web browser. Consider reducing the size of the datasheet by using criteria such as a date field, or by using a parameter query.

[Export a report to static HTML format](#)

1. In the [Database window](#), click the name of the report you want to export, and then on the **File** menu, click **Export**.
2. In the **Save As Type** box, click **HTML Documents (*.html; *.htm)**.
3. Click the arrow to the right of the **Save In** box and select the drive or folder to export to.
4. In the **File Name** box, enter the file name.
5. Select the **AutoStart** check box to display the results in your default [Web browser](#).
6. Click **Save**.
7. In the **HTML Template** box of the **HTML Output Options** dialog box, enter the location of an HTML template. If you do not specify an [HTML](#) template file containing navigation tokens, Microsoft Access provides a default navigation scheme.
8. Once you have exported a report to an HTML file, you need to [publish](#) it to make it available on the [World Wide Web](#).

Notes

- Access exports a report as multiple HTML files, with one file per printed page. The file names are created by using the object name and a suffix; for example, Products.html, ProductsPage2.html, ProductPage3.html, and so on.
- If the report contains a [parameter query](#), Access first prompts you for the parameter values, then exports the results.
- Fields with a Hyperlink data type are exported as hyperlink addresses by using <A HREF> HTML [tags](#).
- Controls and features of a report, including subreports, are supported except

for lines, rectangles, and OLE objects. However, you can use an HTML template file to include images in a [report header](#) or [report footer](#).

- Although a default [theme](#) is not applied when you export the report, you can apply a theme by creating a data access page from an existing Web page (in this case, the static HTML file you just created), applying a theme in page Design view, and then saving the page.
- Access provides several ways to display report information in a Web browser in addition to static HTML reports. You can display a [report snapshot](#) on a Web page, and you can create a [grouped data access page](#). However, you cannot export a report to a server-generated HTML format.

[Export to XML format](#)

When you export to XML format, you can also export the presentation stylesheet (XSL) and create a Web page that displays the XML data formatted according to the stylesheet.

[Export a table, query, form, or report to XML data with the presentation stylesheet](#)

1. In the [Database window](#), click the name of the table, query, form, or report that you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click **XML(*.xml)**.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File name** box, enter a name for the file, and then click **Export**.
5. In the **Export XML** dialog box, check the selections to export the data and the presentation style sheet.
6. Click **More options** to set further options to export static or live data and the presentation style sheet.

To Microsoft Windows SharePoint Services

[Export a table or query to a Windows SharePoint Services list](#)

1. In the [Database window](#), click the name of the table or query you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click **Windows SharePoint Services()**.
3. This opens the Export to Windows SharePoint Services Wizard. Follow the

directions in the wizard to complete the export.

To Microsoft Word or other text format

[Export a datasheet to a delimited or fixed-width text file](#)

1. In the [Database window](#), click the name of the table, query, view, or [stored procedure](#) you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click **Text Files (*.txt; *.csv; *.tab; *.asc)**.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File Name** box, enter a name for the file (or use the suggested name), and then click **Export**.

Microsoft Access starts the Export Text Wizard.

5. Follow the directions in the dialog boxes. Click **Advanced** to create or use an [import/export specification](#).

[Save an object's output as a Rich Text Format file](#)

1. In the [Database window](#), click the name of the object you want to save. To save a selection of a datasheet, open the datasheet and select the portion of the datasheet before continuing.
2. On the **File** menu, click **Export**.
3. In the **Save as type** box, click **Rich Text Format (*.rtf)**.
4. Click the arrow to the right of the **Save in** box, and select the drive or folder to save to.
5. In the **File name** box, enter a name for the file (or use the suggested name).
6. Do one of the following:
 - If you are saving a datasheet, click **Export All** to save the entire datasheet or click the arrow to the right of the **Export All** box, and select **Save Selection** if you selected a portion of the datasheet in step 1.
 - For all other database objects, click **Export**.

[Load the output of a table, query, form, or report into Microsoft Word](#)

1. In the [Database window](#), click the name of the datasheet, form, or report

you want to save and load into Microsoft Word. To save a selection of a datasheet, open the datasheet and select the portion of the datasheet before continuing.

2. On the **Tools** menu, point to **Office Links**, and click **Publish It with Microsoft Word**.

[Save a datasheet to an encoded format](#)

1. On the **File** menu, click **Export**.
2. In the **Save as type** box, click **Text Files (*.txt; *.csv; *.tab; *.asc)**.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File name** box, enter a name for the file (or use the suggested name).
5. Select the **Save formatted** check box.
6. In the **Encode <objectname> As** dialog box, select the method of encoding as **Windows (default)**, **MS-DOS**, **Unicode**, or **Unicode (UTF-8)**.

To XML

[Export to XML](#)

In Microsoft Access, you can output data to [Extensible Markup Language \(XML\)](#) by exporting to an XML document. You can also export a schema to specify the structure of the XML document as well as a stylesheet to describe the presentation of the data.

1. In the [Database window](#), click the name of the table, query, form, or report that you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click **XML(*.xml)**.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File name** box, enter a name for the file, and then click **Export**.
5. In the **Export XML** dialog box, choose to export one or more of the following: the data, the schema, and the presentation style sheet.
6. Click **More options** to set further options to export static or live data, the schema and the presentation style sheet.

To another database

[Export a table or query to a Paradox or dBASE file](#)

Note Microsoft has created updated dBASE and Paradox ISAM drivers for Microsoft Jet 4.0 that do not require the installation of the Borland Database Engine (BDE) to provide full read/write access to dBASE and Paradox files. The default Jet 4.0-based dBase ISAM drivers that ship with Microsoft Data Access Component (MDAC) 2.1 and later, allow read-only access to dBASE and Paradox files unless the BDE is installed. To obtain these ISAM drivers, you must contact Microsoft Technical Support and request that they be sent to you.

1. In the [Database window](#), click the name of the table or query ([Microsoft Access database](#) only) you want to export, and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click the Paradox or dBASE database format you want.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File name** box, enter a name for the file (or use the suggested name), and then click **Export**.

[Export a table or query to an SQL database table or other ODBC data source](#)

Note You can only [export](#) a table or query to an [SQL database](#) or other ODBC [data source](#) if you have installed the appropriate driver.

1. In the [Database window](#), click the name of the table or query ([Microsoft Access database](#) only) you want to export, and then on the **File** menu, click **Export**.
2. In the **Export Object Type 'Object name' To** dialog box, in the **Save As Type** box, click **ODBC Databases ()**.
3. In the **Export** dialog box, enter a name for the file (or use the suggested name), and then click **OK**.

The **Select Data Source** dialog box is displayed; it lists the defined data sources for any [ODBC drivers](#) installed on your computer.

4. Click either the **File Data Source** or **Machine Data Source** tab, and then double-click the ODBC data source that you want to export to.

To define a new data source for any installed ODBC driver, click **New**, and then follow the instructions in the **Create New Data Source** dialog box and the dialog boxes that follow it before continuing.

Particular ODBC data sources have different requirements for accessing their data. For example, most require you to enter a logon ID and password. You might need to get this information from your system administrator. Enter the appropriate information in the dialog box, and then click **OK**.

Microsoft Access connects to the ODBC data source and creates the new table.

Note After you export a table from a Microsoft Access database to an SQL database, you can use an SQL [pass-through query](#) to add an index to the table.

[Show All](#)

Merge tables or queries into Word mail merge files

Note You can also begin the operation of merging data into form letters from within Microsoft Word. For further assistance on this, see Word Help.

 [Export a table or query to a Microsoft Word mail merge data source file](#)

1. In the [Database window](#), click the name of the table or [query](#) you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save As Type** box, click **Microsoft Word Merge (*.txt)**.
3. Click the arrow to the right of the **Save In** box, and select the drive or folder to export to.
4. In the **File Name** box, enter the file name, and then click **Export**.

Microsoft Access creates the data source containing the field names and all the data from your table.

Note In a Word mail merge data source, the first record in the file contains the field names and is called the header row. All succeeding records are the data rows. The field names in the header record must match the field names in the main document. If they don't match, edit the field names either in the data source (you can open the data source in Microsoft Word) or in the main document so they do match.

 [Merge data from a table or query by using the Microsoft Word Mail Merge Wizard](#)

1. In the [Database window](#), click the name of the table or [query](#) you want to [export](#), and then on the **Tools** menu, point to **Office Links** and click **Merge It With Microsoft Word**.
2. Follow the instructions in the **Microsoft Word Mail Merge Wizard**.
3. In the Microsoft Word window, click **Insert Word Field** to insert the desired fields into the document.

In Microsoft Word, if you do not see the **Insert Word Field** icon on the toolbar in Word, point to **Toolbars** on the **View** menu, and then click **Mail Merge**.

[Show All](#)

Set options and send objects in electronic mail

[Add a digital signature to an outgoing message](#)

Note This feature applies only if both you and the recipients of your message use Microsoft Exchange Server. Your system administrator must have set up mail security on your computers. For more information, see Help for Microsoft Exchange Server.

1. Display the message header in Microsoft Outlook. For more information on displaying the message header, see Help in Microsoft Outlook.
2. Click **Options**.
3. Under **Security**, click **Add digital signature to outgoing messages**.

[Add a digital signature to all outgoing messages](#)

Note This feature applies only if both you and the recipients of your message use Microsoft Exchange Server. Your system administrator must have set up mail security on your computers. For more information, see Help for Microsoft Exchange Server.

1. Close any open mail in Microsoft Outlook.
2. Click **Options**.
3. Under **Security**, click **Add digital signature to outgoing messages**.

[Have replies to your message sent to another user](#)

1. Display the message header.
2. Under the **Options** menu item, click **Options**.
3. Under **Delivery options**, select the **Have replies sent to** check box.
4. Type the name of another user, or click **Select Names** to select from a list.

Note You cannot have replies sent to a personal distribution list.

[Manually check names before sending a message](#)

Before you send the output of a [database object](#), or a [data access page](#) as an e-mail message, recipient names you type in the **To**, **Cc**, and **Bcc** boxes are automatically checked against names in the Address Book. If an exact match is found, the name is underlined. If multiple names are found that match the name you type, the **Check Names** dialog box is displayed so that you can select the correct name.

1. Display the message header.
2. Enter recipient names in the **To** and **Cc** boxes. Separate names with a semicolon (;).

To select recipient names from a list, click **To** or **Cc**.

3. Click **Check Names** .

[Tip](#)

You can also press CTRL+K.

[Choose which e-mail account to use to send a message](#)

Note This procedure applies only if you installed Microsoft Outlook in Internet Only mode. Look in Outlook Help for more information on Outlook installation options.

1. Display the message header.
2. Click **Options**.
3. In the **Send message using** box, click the account you want to use, and then click **Close**.

[Send a database object's output as an attachment to an e-mail message](#)

You can attach a database object in an e-mail message in one of the following output formats: [HTML](#) (.htm, .html), Microsoft Excel (.xls), Microsoft MS-DOS text (.txt), Rich Text Format (.rtf), [data access page](#) (.htm, .html) in the case of a data access page, or [report snapshot](#) (.snp) in the case of a report. You can only send modules in MS-DOS text (.txt) output format. You cannot send [macros](#) or,

in the case of [Microsoft Access projects](#), database diagrams.

1. In the Database window, click the type of database object you want under **Objects**, and then click the name of the database object.



If you want to send a selection of a datasheet, open the datasheet and select the portion of the datasheet before continuing.

2. On the **File** menu, point to **Send To**, and then click **Mail Recipient (as Attachment)**.
3. In the **Send** dialog box, click the file format you want, and then click **OK**.
4. Complete the e-mail message. Enter recipient names and distribution lists in the **To** and **Cc** boxes, separate names and lists with a semicolon (;), type your message, and set any additional options you want.
5. When you are ready to mail the message, click **Send** on the message header.

Note Although you cannot attach a database object in either Microsoft Internet Information Services (.htx, .idc) or Microsoft Active Server Pages (.asp) output formats, you can [export](#) the object to the desired output format and then attach the files in a mail program, such as Microsoft Outlook.



You can send a data access page as the body of an e-mail message, or as an attachment. Sending a page as the body of an e-mail message is more useful because supporting files, such as bullets, background textures, and graphics, will be displayed when the recipient views the page.

Security Note Sending the HTML source of a data access page as the body or an attachment of an e-mail poses a security risk. The HTML source can be intercepted and modified to run malicious scripts that can inflict harmful effects on your computer. It is more secure to send a URL to the data access page in an email.

1. In the Database window, click **Pages** under **Objects**, and then click the name of the page.

2. On the **File** menu, point to **Send To**, and then click **Mail Recipient**.
3. Do one of the following:
 - If you receive the message **Create a message that points to your saved data access page?**, click **Yes**.
 - If you receive the message **Create a message that contains a copy of your data access page?**, click **Cancel**, then save the page using a [universal naming convention](#) (UNC) path to a [Web server](#) or [public folder](#) and begin this procedure again.
4. Complete the e-mail message. Access displays the page in [Design view](#) so that you can make changes to the page, such as adding text or notes for the mail recipient, before you send it.
5. When you are ready to mail the message, click **Send a Copy** on the message header.

[Show All](#)

Data formats you can export to

Application	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002, Access 2003
Microsoft Access project	9.0/2000, 10.0/Access 2002, Access 2003
dBASE	III, IV, and 5; and 7 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Microsoft Excel	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Excel 2002, and Excel 2003
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Microsoft Windows SharePoint Services	2.0
Microsoft Word, Rich Text Format	All
Lotus 1-2-3	.wj2, .wk1 and .wk3 formats
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML and IDC/HTX	1.0 (if a list) 2.0, 3.x, 4.x (if a table or list)
Microsoft Active Server Pages	All
XML Documents	All
SQL tables, Microsoft Visual FoxPro, and programs and databases that support the ODBC protocol	Visual FoxPro 3.0, 5.0, and 6.x For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

[Show All](#)

Troubleshoot exporting in Access

[File extensions don't show up beside the file names in the **Save As Type** dialog box.](#)

You have not selected the **Show hidden files and folders** option in Microsoft Windows Explorer. See Windows Help for information on selecting this option.

[I'm not able to link my database objects to Microsoft IIS or Microsoft ASP files.](#)

Although you cannot link a database object to either Microsoft Internet Information Services (.htx, .idc) or Microsoft Active Server Pages (.asp) output formats, you can [export](#) the object to the desired output format and then attach the file in a mail program, such as Microsoft Outlook.

[Show All](#)

Export Access data as XML

In Microsoft Access, you can output data to [Extensible Markup Language \(XML\)](#) by exporting to an XML document. You can also export a schema to specify the structure of the XML document as well as a stylesheet to describe the presentation of the data.

1. In the [Database window](#), click the name of the table, query, form, or report that you want to [export](#), and then on the **File** menu, click **Export**.
2. In the **Save as type** box, click **XML(*.xml)**.
3. Click the arrow to the right of the **Save in** box, and select the drive or folder to export to.
4. In the **File name** box, enter a name for the file, and then click **Export**.
5. In the **Export XML** dialog box, do one or more of the following:
 - To export the data to an XML document, select **Data (XML)**.
 - To export the schema to an XML file, select **Schema of the data**.
 - To format your table or query data by using an Extensible Style Language file, select **Presentation of your data (XSL)**.
6. [If you are exporting live data or if you want to specify further options for exporting your static data, click **More options**](#)

The **Export XML** dialog expands to let you

- export static data and apply a transform, include related tables or queries, or apply a filter or sorting criteria,
- export live data,
- specify options to export a schema, or
- specify options to export a presentation stylesheet.

Do one or more of the following:

[Export static or live data to XML](#)

1. Select the **Data** tab.

Note The **Export data** check box shows the selection made in step 5.

2. Enter a destination for the exported files in the **Export location** dialog box or use the **Browse** button. The path can be to a hard disk drive or a [universal naming convention \(UNC\)](#) path on a server.
3. To select an encoding format other than the default, click the **Encoding** drop-down box and make a selection.
4. [If you are exporting static data in a table, query, or a report or form that has a table or a query as a static data source](#)

- The tree under **Data to Export** displays the database object you are exporting and any related tables or queries. To include any related tables or queries in the exported data, select them in the tree.

Note You cannot exclude the database object that you have selected to export in the Database window.

- Under **Records to Export**, you can choose to export all records in the selected database objects, apply a predefined filter and export only the filtered records, or export only the current record.

Note You can only apply a filter if the filter has already been defined for the table or query.

- Select **Apply Existing Sort** if you want to sort the exported data according to any predefined sorting criteria.
- [Click **Transform** to specify any transform that you want to apply to the exported data.](#)

In the **Export Transforms** dialog box, do one of the following.

- In the displayed list of transforms, select a transform and click **OK**.

Note Selecting **(none)** and clicking **OK** is the same as not applying any transform.

- If the displayed list does not contain the necessary transform, add the transform file by clicking **Add**. This opens the **Add**

New Transforms dialog box which allows you to add a file with the extension xsl or xslt. After adding the transform, select it and click **OK**.

Note If the XML data references any transforms, the **Export Transforms** dialog box also lists these transforms. These transforms and the item (**none**) cannot be renamed or removed.

5. [If you are exporting live data in a table, query, stored procedure, function, or a form or report in a Microsoft Access Project \(ADP\)](#)

1. Select **Live Data**. The database object you are exporting is connected to a live data source on a SQL server.

Note The **Live Data** option in the Export XML dialog is disabled if you are trying to export a form or a report that is bound to a stored procedure. However, this can be accomplished programmatically through the Access object model.

2. Specify the **Virtual Directory** for the live data source. This virtual directory is the HTTP address of the SQL server data source. It defaults to http://<machine name of the SQL server>/<database>. For more information on how to set up a virtual directory, see "Using IIS Virtual Directory Management for SQL Server Utility" on the Microsoft Developer Network (MSDN). For an example of how to set up a virtual directory, see "Creating the nwind Virtual Directory" on MSDN.

[Export schema to XML](#)

1. Select the **Schema** tab.

Note The **Export schema** check box shows the selection made in step 5.

2. To include additional information on the underlying structure of the database such as the primary key and index information, select the **Include primary key and index information** check box.
3. To embed the schema directly into the XML data document, select the

Embed schema in exported XML data document option.

To create a schema file which is separate from the data file, select the **Create separate schema document** option, and enter a valid destination for the file or use the default.

[Present your data](#)

1. Select the **Presentation** tab.

Note The **Presentation format** drop-down box shows the selection made in step 5.

2. Do one of the following:
 - To run the exported file from a client machine, select the **Client (HTML)** option under the **Run from** selection. This option generates an HTML file on the client machine which contains scripting code to merge the presentation (XSL) file and the data (XML) file. This option has the advantage of indirectly referencing the presentation file from the HTML file instead of embedding the presentation information in the data. Therefore, either the data (XML) file or presentation (XSL) file can be changed without having to rerun the original document.
 - To run the exported file from a server, select the **Server (ASP)** option under the **Run from** selection. This option generates an ASP file that merges the presentation (XSL) information with the data (XML) and sends the resulting HTML file to the client machine for display.
3. When you are exporting forms or reports, in order for images to appear correctly in an XML presentation, they must be located external to the XSL document which contains internal references to the files. Do one of the following:
 - Select the **Put images in** option and the file location.
 - Select the **Don't include images** option to exclude any document images from the XML version.
4. Enter a destination for the exported presentation file in the **Export location** dialog box, or use the **Browse** button. The destination path can be to a hard disk drive or a [universal naming convention](#) (UNC) path on a server.

7. If you do not need to specify further options, click **OK** to complete the export operation.

About XSL transformation files

One of the advantages of using XML as a data interchange standard is the ease of transforming XML data documents into other formats and structures, such as HTML files. The two languages that are used to define transformations are XML-based languages called Extensible Stylesheet Language (XSL) and Extensible Stylesheet Language: Transformations (XSLT). XSLT is a subset of XSL. XSL is used to define the formatting of XML documents, and XSLT defines the structure of the data.

You can specify a transformation file when you import data from or export data to XML. When you import XML data, you can transform the data as soon as the data is imported— before you create a new table or append data to an existing one. When you export data to XML, you can select a transformation that is applied following the export.

When you export data to XML, you can also choose to export a presentation XSL file. The file is generated based on presentation information in the Access form, report, or data sheet. You can then modify this file to further change the formatting of the exported XML data.

[Show All](#)

About importing and linking data and database objects

Microsoft Access provides two choices for using data from an external data source. You can:

- Import the data into a new Microsoft Access table, which is a way to convert data from a different format and copy it into Microsoft Access. You can also import database objects into another [Microsoft Access database](#) or [Microsoft Access project](#).
- Link to the data, which is a way to connect to data from another application without importing it so that you can view and edit the data in both the original application and in an [Access file](#). (In previous versions of Microsoft Access, this process was referred to as attaching.)

You can import or link data from a variety of supported databases, programs, and file formats.

[Should I import or link a table?](#)

Reasons to import data into a table

If you know that you will use your data only in Microsoft Access, you should import it. Microsoft Access generally works faster with its own tables, and, if you need to, you can modify the imported table to meet your needs just like any other table that is created in Microsoft Access.

Reasons to link data from an external table

You might want to link to tables from other Microsoft Access databases if, for example, you want to use a table from another Microsoft Access database that is shared on a network. This is particularly useful if you want to store all of your tables in one database on a network server, while keeping forms, reports, and other objects in a separate database that is copied among users of the shared database. You can easily split an existing database into two databases, called front-end and back-end databases.

If the data that you want to use in your Access database is also being updated by a program other than Microsoft Access, you should link it. Using this approach, the current methods of updating, managing, and sharing the data can remain in place, and you can use Microsoft Access to work with the data as well. For example, you can create queries, forms, and reports that use the external data, combine external data with the data in Microsoft Access tables, and even view and edit the external data while others are using it in the original program.

You might also want to link to SQL Server tables from your Access project. You can create a linked table view by using the Linked Table Wizard. These views can then be used in forms, reports, data access pages, and queries just like any other table.



[What happens when importing data.](#)

Importing data creates a copy of the information in a new table in your Access database or Access project. The source table or file is not altered in this process.

When importing data, you can't append data to existing tables (except when importing spreadsheet or text files). However, once you have imported a table, in an Access database you can perform an [append query](#) or in an Access project you can use a [stored procedure](#) or an append query to add the table's data to another table.

You can also import database objects other than tables, such as forms or reports, from another Access database or Access project.

[What happens when linking data.](#)

Linking data enables you to read and, in most cases, update data in the external data source without importing. The external data source's format is not altered so that you can continue to use the file with the program that originally created it, but you can add, delete, or edit its data by using Microsoft Access as well.

Note You cannot update data that is linked to an Excel worksheet.

Microsoft Access uses different icons to represent linked tables and tables that are stored in the current database. If you delete the icon for a linked table, you delete the link to the table, not the external table itself.



- 1 Linked DBASE table
- 2 Linked Paradox table
- 3 Linked Access table

In Microsoft Access projects, a view is created that references the linked table. The view can then be referenced in forms, reports, data access pages, and queries. The ability to insert, update, or delete records from the resulting view depends on the capabilities of the data source selected.



[Import and link data from ODBC data sources.](#)

You can import or link data from [ODBC databases](#), such as Microsoft SQL Server and Visual FoxPro, as well as other programs that provide drivers compliant with [ODBC Level 1](#) to access their data files. To do this, you must be connected to the appropriate [ODBC data source](#). To connect to an ODBC data source, you must have the correct [ODBC driver](#) installed and a data source name defined.

When you import a table into an Access project, Microsoft Access imports the table data, data definitions, and [primary key](#) but not its other properties (including constraints, [relationships](#), and indexes).

[Unsupported programs.](#)

If you have a program whose data is not stored in one of the supported external database or file formats, but the program can export, convert, or save its data as one of these formats, then you can import that data. Additionally, when you import or link an external database format, such as dBASE or Paradox, Access usually preserves indexes. For example, you can't import or link a Microsoft Works database directly, but you can export the data from Microsoft Works to a dBASE IV (.dbf) database file and then import that data into Microsoft Access to preserve field names and indexes.

In general, most unsupported programs, even those on different operating systems, can export data to a [delimited](#) or [fixed-width text file](#), which you can then import or link from Microsoft Access, although you won't preserve indexes.

[Automating import operations.](#)

To make frequent import operations more convenient, you can automate them by creating a macro or creating a [Microsoft Visual Basic for Applications](#) procedure. This is useful, for example, when you import data on a regular schedule or you have unusual or complex requirements for importing data.

[Linked table properties.](#)

When you open a linked table in [Design view](#), it looks much like a regular Microsoft Access table. Although you can't change the way the linked table and its fields are defined in the external database, you can set the properties that control the way the fields appear in Microsoft Access. Changes that you make to

properties for linked tables affect only how Microsoft Access handles and displays data from the linked table; no changes are made to the source table. Field properties that you can set for linked tables are listed in the following table.

Property	Effect
Format	Controls how data is displayed in a field
DecimalPlaces	Controls the number of decimal places displayed
InputMask	Creates a data input mask with separator characters and blanks to fill in
Caption	Changes the name used for a field's datasheet column heading, and specifies a default name to be used as a label when adding a field to a form

You can't change other field properties for linked tables. However, to help make data entry more efficient and reliable, you can create forms that you use to add or edit data in your linked tables, and set properties for [controls](#) that are bound to fields from your linked tables. For example, you might want to set the **DefaultValue**, **ValidationRule**, and **ValidationText** properties for controls in those forms.

If you link two tables from the same Access database, then any [relationship](#) that was established between the tables in the other database remains in effect.

If you link tables from another Microsoft Access database, the tables use the property settings from the database they are stored in. For example, if a table has validation rules in the original database, data that you enter in the linked table must follow these rules as well. If you need to change these properties, you must open the table in the database it is stored in.

You might want to rename linked tables. Because Microsoft Access table names can contain spaces and up to 64 characters, you might want to give a linked table a more descriptive name after you link it. For example, if you link a dBASE table named SLSDATA, you could rename the linked table "Sales Data (from dBASE)." Note that this won't rename the table itself, just the name Microsoft Access uses to refer to the link to that table.

[About import/export specifications and schema.ini files.](#)

An import/export specification contains information, such as file format, date order, or number formats, that Microsoft Access uses to import or export a fixed-width or [delimited text file](#). An import/export specification is stored with the default name *Filename_ImportSpec* or *Filename_ExportSpec* in the database that you import to or export from.

You create an import/export specification by using either the Import Text Wizard or the Export Text Wizard. Use an import/export specification when you want to repeatedly import to the same table, export to the same file, or automate the import or export process.

Alternatively, you can use a schema.ini file in a Microsoft Visual Basic program to provide even more control over data in the text file, such as specifying special currency formats or handling floating point data types. A schema.ini file is a text file containing entries that override default text driver settings in the Microsoft Windows registry. You store a schema.ini in the same folder as the imported or exported text file, and it must always be named schema.ini.

[Show All](#)

Data sources you can import or link

Data source	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002, Access 2003
Microsoft Access project	9.0/2000, 10.0/Access 2002, Access 2003
dBASE	III, IV, and 5; and 7 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Excel spreadsheets	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Excel 2002, and Excel 2003
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Windows SharePoint Services 2.0	
Lotus 1-2-3 spreadsheets (Linking is read-only)	.wks, .wk1, .wk3, and .wk4
Microsoft Exchange	All versions
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML	1.0 (if a list) 2.0, 3.x (if a table or list)
XML Documents	All versions (import only)
SQL tables, Microsoft Visual FoxPro, and data from other programs and databases that support the ODBC protocol	Visual FoxPro 2.x, 3.0, 5.0, and 6.x (import only) For an up-to-date list of supported ODBC drivers , see the Microsoft Knowledge Base.

[Show All](#)

Import or link data and objects

Important If you link to a file on a local area network, make sure to use a [universal naming convention \(UNC\)](#) path, instead of relying on the drive letter of a mapped network drive in Microsoft Windows Explorer. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for Microsoft Access to locate the data source that contains the linked table.

From another Access file



[Import or link tables from another Microsoft Access file](#)

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To [import](#) tables, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link tables, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, make sure that **Microsoft Access (*.mdb; *.adp; *.mda; *.mde; *.ade)** is selected.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the Access file that you want to import or link is located, and then double-click the database's icon.
5. In the **Import Objects** (or **Link Tables**) dialog box, click each table that you want to import or link.

If you're importing, and you want to import just the selected tables' definitions (not the data they contain), click **Options**, and then under **Import Tables**, click **Definition Only**.

Note If you import a table that is already linked, then Access does not import the data; instead, it links the table to its data source (in effect, copies the link information).

From a text file

[Import or link a delimited or fixed-width text file](#)

Important Before you [import](#) or link data from a [delimited text file](#) or [fixed-width text file](#), make sure that the file has the same type of data in each field and the same fields in every row.

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To import data, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link data, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select **Text Files (*.txt; *.csv; *.tab; *.asc)**.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the file is located, and then double-click its icon.
5. Follow the directions in the **Import Text Wizard** dialog boxes. Click **Advanced** to create or use an [import/export specification](#).

To cancel importing, press CTRL+BREAK.

Notes

- You can append the data to an existing table as long as the first row of your text file matches the table's field names.
- If all the records in a fixed-width text file are the same length, an embedded row separator (such as a carriage return and a linefeed) in the middle of a record will not cause unexpected results. However, if the records aren't all the same length, Microsoft Access will treat the embedded row separator as the end of the record.

From a spreadsheet

[Import or link data from a spreadsheet](#)

Before you proceed, make sure that the data in the spreadsheet is arranged in an appropriate tabular format, and the spreadsheet has the same type of data in each field (column) and the same fields in every row.

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To [import](#) a spreadsheet, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link a spreadsheet, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select **Microsoft Excel (*.xls)**.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the spreadsheet file is located, and then double-click its icon.
5. Follow the directions in the **Import Spreadsheet Wizard** dialog boxes. If you are importing from a Microsoft Excel version 5.0 or later workbook, then you can import from one worksheet within a workbook. You cannot import from any other multiple-spreadsheet files, such as Microsoft Excel version 4.0 workbooks. To import from these files, you must first save each spreadsheet as an individual file.

Notes

- You can import or link all the data from a spreadsheet, or just the data from a named range of cells. Although you normally create a new table in Microsoft Access for the data, you can also append the data to an existing table as long as your spreadsheet column headings match the table's field names.
- Access attempts to assign the appropriate data type to imported fields, but you should check your fields to make sure that they are set to the data type you want. For example, in an Access database, a phone number or postal code field might be imported as a Number field, but should be changed to a Text field in Microsoft Access because it is unlikely that you will perform any calculations on these types of fields. You should also check and set field properties, such as formatting, as necessary.

From a Web page

 [Import or link \(read-only\) HTML tables and lists](#)

1. Open a database, or switch to the Database window for the open database.
2. To import HTML tables or lists, on the **File** menu, point to **Get External Data**, and then click **Import**.

To link HTML tables or lists, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.

3. In the **Import** or **Link** dialog box, in the **Files of type** box, click **HTML Documents (*.html; *.htm)**.
4. Click the arrow to the right of the **Look in** box.
5. Select the drive and folder where the HTML file you want to import or link is located, and then double-click the file name.
6. Follow the instructions for the Import HTML Wizard or the Link HTML Wizard. Click the **Advanced** button if you want to edit an [import/export specification](#) or specify different file and field formats.
7. If your HTML file contains more than one table or list, repeat steps 1 through 6 for each table or list you want to import or link.

Notes

- By default, Access converts a hyperlink address to a Hyperlink data type column, but only if all values in a table column or list contain a hyperlink address defined by an <A HREF> HTML tag. You can change the data type when using the Import HTML Wizard or the Link HTML Wizard.
- A table embedded within a table cell in an HTML file is treated as a separate table when you import or link. A list embedded in a table cell is treated as the contents of a cell, and each item in the list is delimited with the carriage return/line feed characters.
- For data that spans rows or columns in an HTML table, Microsoft Access 2000 (and later versions) duplicates the data in each cell, whereas Microsoft Excel 2000 (and later versions) stores the data in the first or upper-left cell, and then leaves other cells blank.

From Windows SharePoint Services

 [Import or link data from Microsoft Windows SharePoint Services lists](#)

You can use the Import from Windows SharePoint Services Wizard to import tables from Windows SharePoint Services lists, or use the Link to Windows SharePoint Services Wizard to create and link tables to Windows SharePoint Services lists.

1. Open a database, or switch to the [Database window](#) for the open database.

2. Do one of the following:
 - To [import](#) data, on the **File** menu, point to **Get External Data**, and then click **Import**. In the **Import** dialog box, in the **Files of type** dialog box, select **Windows SharePoint Services()**.
 - To link data, on the **File** menu, point to **Get External Data**, and then click **Link Tables**. In the **Link** dialog box, in the **Files of type** dialog box, select **Windows SharePoint Services()**.
3. Follow the directions in the **Import from Windows SharePoint Services Wizard** or the **Link to Windows SharePoint Services Wizard**.

From SQL or another ODBC data source

 [Import or link SQL database tables or data from other ODBC data sources](#)

Note You can only [import](#) or link an [SQL](#) or other [ODBC data source](#) if you have installed the appropriate driver.

1. Open an Access file, or switch to the [Database window](#) for the open Access file.
2. Do one of the following:
 - To import tables, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link tables, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select **ODBC Databases()**.

The **Select Data Source** dialog box lists the defined data sources for any [ODBC drivers](#) that are installed on your computer.

4. Click either the **File Data Source** or **Machine Data Source** tab, and then double-click the ODBC data source that you want to import.

To define a new data source for any installed ODBC driver, click **New**, and then follow the instructions in the **Create New Data Source** dialog box and the dialog boxes that follow it before continuing.

5. If the ODBC data source that you selected requires you to log on, type your logon ID and password (additional information might also be required), and then click **OK**.

Microsoft Access connects to the ODBC data source and displays the list of tables that you can import or link.

6. If you're linking a table, do not select the **Save password** check box if you want to keep the database secure. This will force users to type the logon ID and password every time they open the table in each new session with Microsoft Access. Your [SQL database](#) administrator can also choose to disable this check box, requiring all users to type the logon ID and password each time they connect to the SQL database.
7. Click each table that you want to import or link, and then click **OK**. If you're linking a table and it doesn't have an index that uniquely identifies each record, then Microsoft Access displays a list of the fields in the linked table. Click a field or a combination of fields that will uniquely identify each record, and then click **OK**.

From a mail program

[Import or link data from Microsoft Exchange or Microsoft Outlook](#)

You can use the Import from Exchange/Outlook Wizard or the Link to Exchange/Outlook Wizard to import or link to your Microsoft Outlook Contacts folder and then create form letters and mailing labels by merging the data using the Microsoft Word Mail Merge Wizard. You must have Microsoft Outlook, Outlook Express, or Microsoft Exchange installed on your computer to run these wizards.

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To [import](#) data, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link data, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select either

Outlook() or **Exchange()**.

4. Follow the directions in the **Import from Exchange/Outlook Wizard** or the **Link to Exchange/Outlook Wizard**.

From another database

[Import a Microsoft Works database](#)

You can't [import](#) a Microsoft Works database (.wdb) directly into Microsoft Access. You must first use Microsoft Works to save the file to one of the file formats that Microsoft Access can import.

1. Open the database by using Microsoft Works.
2. On the **File** menu, click **Save as**.
3. In the **Save as type** box, select **dBASE IV (*.dbf)**, and then click **OK**.
4. Import the dBASE IV .dbf file that you created in step 3 into Microsoft Access.

[Import or link dBASE files](#)

Microsoft has created dBASE ISAM drivers for Microsoft Jet 4.0 that do not require the installation of the Borland Database Engine (BDE) to provide full read/write access to dBASE files. The default Jet 4.0-based dBase ISAM drivers that ship with Microsoft Data Access Component (MDAC) 2.1 and later, allow read-only access to dBASE files unless the BDE is installed. To obtain these ISAM drivers, you must contact Microsoft Technical Support and request that they be sent to you.

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To [import](#) files, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link files, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select one of the dBASE file types.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the .dbf file is located, and then double-click its icon.

- If importing, Microsoft Access creates a table named after the file that you selected, and then imports the data from the .dbf file.
- If linking, Microsoft Access displays a dialog box where you can associate dBASE index files. Do the following:
 1. Double-click each dBASE index (.ndx or .mdx) file that you want to use, and then click **Close**. If there are no indexes, click **Cancel** to continue.
 2. In the **Select Unique Record Identifier** dialog box, select an index that uniquely identifies each record. This index must contain no duplicate values, or Microsoft Access might not be able to properly update data in queries with joins.
- 5. To import or link another dBASE file, repeat step 4.

Note After importing or linking a .dbf or .dbc file, you can set field properties for the table. If you import a .dbf or .dbc file, you might also want to set a [primary key](#) for the table.

[Import or link Paradox tables](#)

Microsoft has created Paradox ISAM drivers for Microsoft Jet 4.0 that do not require the installation of the Borland Database Engine (BDE) to provide full read/write access to Paradox files. The default Jet 4.0-based Paradox ISAM drivers that ship with Microsoft Data Access Component (MDAC) 2.1 and later, allow read-only access to Paradox files unless the BDE is installed. To obtain these ISAM drivers, you must contact Microsoft Technical Support, and request that they be sent to you.

1. Open a database, or switch to the [Database window](#) for the open database.
2. Do one of the following:
 - To [import](#) tables, on the **File** menu, point to **Get External Data**, and then click **Import**.
 - To link tables, on the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the **Import** (or **Link**) dialog box, in the **Files of type** box, select **Paradox (*.db*)**.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the .db file is located, and then double-click its icon.
5. If the Paradox table you select is encrypted, Microsoft Access prompts you for the password. Type the password for the Paradox table, and then click

OK.

6. If you want to import or link another Paradox table, repeat steps 2 through 5.

From a data access page

[Import a data access page from a Microsoft Access file](#)

When you [import](#) a [data access page](#), you are only importing the link to the corresponding HTML file. The HTML file remains in the same location, which means that you have two links pointing to the same corresponding HTML file. If you don't want this, save one of the linked pages under a different name (using the **Save As** command on the **File** menu), type a different file location for the new corresponding HTML file in the **Save As** dialog box, and then delete the linked page that you just saved.

1. Open a database, or switch to the [Database window](#) for the open database.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. In the **Import** dialog box, in the **Files of type** box, make sure that **Microsoft Access (*.mdb; *.adp; *.mda; *.mde; *.ade)** is selected.
4. Click the arrow to the right of the **Look in** box, select the drive and folder where the Access database (.mdb) or Access project (.adp) is located, and then double-click the Access database's or Access project's icon.
5. Under the **Pages** tab in the **Import Objects** dialog box, click each data access page that you want to import.

From data in XML

[Import data and schema from XML](#)

1. Open a database, or switch to the [Database window](#) for the open database.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. In the **Import** dialog box, in the **Files Of Type** box, select **XML Documents (*.xml;*.xsd)**.
4. Click the arrow to the right of the **Look In** box, select the drive and folder where the file is located, and then double-click its icon.
5. The **Import XML** dialog box displays a list of tables contained within the XML document.

Note All of the tables shown in the list will be imported. You cannot select a subset of the XML document for importing.

6. To start importing the file, click **OK** . This completes the import procedure.

Note For most records that cause an error, Microsoft Access creates and adds a row to a table called Import Errors. To view the list of errors that were encountered, open the Import Errors table from the Database window.

7. To set options for importing, select one of the following under **Import Options**:
 - To import just the structure of the table, click **Structure Only**.
 - To import the structure of the table and the data, click **Structure and Data**.
 - To append the data to an existing table, click **Append Data To Existing Table(s)**.

Click **OK** to start importing, or proceed to the next step.

Note For the first two options above, if a table already exists with the same name, a new table is created with a number appended to the name. Otherwise, a new table is created.

8. [To select a transform to apply to the data being imported, click **Transform**.](#)

In the **Import Transforms** dialog box, do one of the following.

- In the displayed list of transforms, select a transform and click **OK**.

Note Selecting **(none)** and clicking **OK** is the same as not applying any transform.

- If the displayed list does not contain the necessary transform, add the transform file by clicking **Add**. This opens the **Add New Transforms** dialog box which allows you to add a file with the extension xsl or xslt. After adding the transform, select it and click **OK**.

Note If the XML data references any transforms, the **Import Transforms** dialog box also lists these transforms. These transforms and the item **(none)** cannot be renamed or removed.

[Show All](#)

About using the MSysConf table to optimize your linked SQL database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Important If you create an MSysConf table, it must be set up properly or you will not be able to connect to the [SQL database](#) table at all.

If you are administering an SQL database that uses Microsoft Access as a front end, then you can create a table in your SQL database named MSysConf to help you control communication between the two applications. The MSysConf table has two potential functions:

- It can disable the feature that enables users to save the logon ID and password for a linked SQL database in the Microsoft Access front end.
- It can optimize how Microsoft Access performs background population of records during idle time by setting the number of rows of data that are retrieved at one time and the number of seconds of delay between each retrieval. Microsoft Access uses a default of 100 records that are returned every 10 seconds if you don't create the MSysConf table.

 [The structure of the MSysConf table](#)

The SQL database table MSysConf should have the following structure.

Column name	Data type	Allows Null?
Config	A data type that corresponds to a 2-byte integer	No
chValue	VARCHAR(255)	Yes
nValue	A data type that corresponds to a 4-byte integer	Yes
Comments	VARCHAR(255)	Yes

Note If the data source you're working with is case-sensitive, then use the table

and column names exactly as shown.

[The data in the MSysConf table](#)

There are three valid records in the MSysConf table. The following table shows what values you should enter in the Config and nValue field. The other columns are reserved for future use, and their contents are ignored.

Config nValue		Meaning
101	0	Don't allow local storage of the logon ID and password in linked tables.
101	1	Allow local storage of the logon ID and password in linked tables.
102	<i>D</i>	<i>D</i> is the delay in seconds between each retrieval.
103	<i>N</i>	<i>N</i> is the number of rows retrieved.

Note Setting a higher delay time decreases network traffic, but increases the amount of time that read-locks are left on data (if the server uses read-locks).

[Show All](#)

View, refresh, or change the file name and path for linked tables (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can link a table only in a [Microsoft Access database](#), not a [Microsoft Access project](#).

[View or refresh links](#)

Use this procedure to view or to refresh links when the structure or location of a linked table has changed. The Linked Table Manager lists the paths to all currently linked tables.

1. Open the database that contains links to tables.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Linked Table Manager**.
3. Select the check box for the tables whose links you want to refresh.
4. Click **OK** to refresh the links.

Microsoft Access confirms a successful refresh or, if the table wasn't found, displays the **Select New Location of <table name>** dialog box in which you can specify the table's new location.

If several selected tables have moved to the new location that you specify, the Linked Table Manager searches that location for all selected tables, and updates all links in one step.

[To change the path for a set of linked tables](#)

1. Open the database that contains links to tables.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Linked Table Manager**.
3. Select the **Always prompt for new location** check box.

4. Select the check box for the tables whose links you want to change, and then click **OK**.
5. In the **Select New Location of** *<table name>* dialog box, specify the new location, click **Open**, and then click **OK**.

Notes

- The Linked Table Manager doesn't move database or table files. For information about moving database or table files to a new location, search Microsoft Windows Help. Once you have moved a database or table, you can use the Linked Table Manager to refresh links.
- The Linked Table Manager can't refresh links to Microsoft Access tables whose names were changed after they were linked. You must delete the current link, and then link such tables again.

[Show All](#)

Create a field to store paths to pictures

1. Create a field in a table and set the field's data type to **Text** in a [Microsoft Access database](#) or **varchar** in a [Microsoft Access project](#).
2. In [Datasheet view](#), type the path to the image (a .bmp, .jpeg, .gif, .dib, .wmf, or .emf file) in the field you created.

If you will be sharing the Access database, you might need to specify a [UNC path](#).

If you will be displaying the pictures in a [data access page](#), you might need to specify a [URL](#) to a valid Web address.

Caution Users of your database or data access page will be able to view the image paths. You might want to make sure that the paths do not include personal information, such as your name or the name of your computer.



You can use relative paths to store pictures. For example, if the [Access file](#) is located at `\\ServerName\ShareName\` and the pictures are stored in the folder `\\ServerName\ShareName\Pictures`, you don't have to specify the entire path such as `\\ServerName\ShareName\Pictures\PictureFileName.gif` for a picture that you will display in a form or report. You only need to store the portion of the path that is different relative to the location of the current Access file — **Pictures\PictureFileName**.

If you want to use relative paths to store the locations of pictures that you will display in data access pages, those paths must be relative to the location of the data access page, not the location of the Access database or Access project.

[Show All](#)

Refresh the data in a linked picture or object

When you link to an object that was created in another application, Microsoft Access creates an [automatic link](#) by default. You can change the type of link to [manual](#), or vice versa. This procedure only applies to linked pictures or objects in [unbound object frames](#) or [bound object frames](#) on a form or report. Background pictures and pictures in [image controls](#) are always updated automatically.

Note This type of link is different from a hyperlink that is used on a form or report to jump to another document, object, or page.

1. Do one of the following:
 - If the object is [unbound](#), click the object in form or report [Design view](#).
 - If the object is [bound](#), open the form in [Form view](#) (or open a datasheet), find the record you want to change, and click the object.
2. On the **Edit** menu, click **OLE/DDE Links**.
3. In the **Links** box, click the link you want to update. To select multiple links, hold down the CTRL key while you click each link.
4. Do one of the following:
 - To refresh the data, click **Update Now**.
 - To specify that the data is refreshed whenever new information becomes available, click **Automatic**.
 - To specify that the data is refreshed only when you press SHIFT+F9 or click **Update Now** in the **Links** dialog box, click **Manual**.

Note When you make any changes to a link for a bound object frame, you are changing the link only for the current record. To make a change in a different record, move to that record and repeat the process.

[Show All](#)

Use custom colors from another application

The colors of an image as it appears in a form or report might not be the same as the colors of that image in the application where it was originally created. To ensure that an image appears in the original colors, you can specify a custom palette for the form or report. The custom palette replaces the Microsoft Access default palette only for that form or report.

1. Open the form or report in [Design view](#).
2. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
3. In the **PaletteSource** property box, type the path and file name of one of the following types of files:
 - .dib (device-independent bitmap file)
 - .pal (Microsoft Windows palette file)
 - .ico (Windows icon file)
 - .bmp (Windows bitmap file)
 - .wmf (Windows metafile)
 - .emf (enhanced metafile)
 - Any other file types supported by graphic filters that you have installed

Note To display pictures and charts in their original colors, the display driver currently loaded in your system must support the palette of the graphic file whose path and file name appear in the **PaletteSource** property box. For example, if you specify **C:\Windows\256Color.bmp**, the currently loaded display driver must support 256 colors.

[Show All](#)

Delete the link to a linked table in an Access database

When you delete a linked table, you are deleting only the information that Microsoft Access uses to open the table, not the table itself. You can link to the same table again later.

1. In the [Database window](#), click **Tables** under **Objects**, and then click the linked table.
2. Press the DELETE key.

Microsoft Access deletes the link and removes the table's name from the list.

[Show All](#)

Troubleshoot importing and linking

In a [Microsoft Access database](#), for most records that cause an error, Microsoft Access adds a row to a table called Import Errors. To view the list of errors that were encountered, open the Import Errors table from the [Database window](#).



[I get errors when I import a spreadsheet or text file.](#)

Here are the most typical reasons for encountering errors:

- In a field, there is data that can't be stored in the data type that Microsoft Access has assigned to that field. For example, you might have accidentally included a text value in a field that should only contain numbers or dates. Or perhaps a row in your text file or spreadsheet contains summary information or extraneous characters.

If you think Microsoft Access assigned the correct data type for this field, edit your text file or spreadsheet to correct errors, and then import again. Otherwise, import again and specify the appropriate data type.

- Microsoft Access has assigned an incorrect data type to a field. Microsoft Access assigns the data type for each field based on the data in the first row it imports. For example, if a field that contains mostly text values has a number in the first row, then, in an Access database, Access assigns the Number data type. In an Access project, Access assigns a data type for numeric data. In both cases, Access cannot import the rest of the records.

Import again and specify the appropriate data type when importing.

- One or more rows in the text file or spreadsheet contain more fields than the first row. For example, the second row in a file might have an extra field delimiter character that is followed by a value that Microsoft Access can't fit in the new table.

Edit your text file or spreadsheet so that each row has the same number of fields, and import again.

- The date fields that are imported from a Microsoft Excel spreadsheet are off by four years. Microsoft Excel for Windows uses the 1900 Date System, in which serial numbers range from 1 to 65,380, which correspond to the dates January 1, 1900 through December 31, 2078. However, Microsoft Excel for the Macintosh uses the 1904 Date System, in which serial numbers range from 0 to 63,918, which correspond to the dates January 1, 1904 through December 31, 2078.

Before importing the data, change the date system for the Microsoft Excel spreadsheet, or, after importing the data, perform an [update query](#) by using the expression [*date field name*] + 1462 to correct the dates.

[Importing a text file is taking a long time.](#)

If importing a text file takes an unexpectedly long time, it might be because many errors are occurring. To cancel importing, press CTRL+BREAK. To view the list of errors that were encountered, open the Import Errors table from the Database window.

[I get errors when I append a spreadsheet or text file.](#)

Appended records must be compatible with the existing table's structure: Each field must have the same data type as the corresponding field in the destination table, and the fields must be in the same order (unless you're using the first row of the file as field names, in which case the field names must match).

If you suspect that the problem is with the data that you're appending, edit your spreadsheet or text file, and then import again. Alternatively, you might need to change the destination table itself. You can reorder the fields, change data types, run an [update query](#) (to reformat or recalculate the data, for example), or run a [make-table query](#) (to split one field into two fields or combine several fields into one field, for example).

Here are the most typical reasons for encountering errors:

- The field names in the spreadsheet or text file do not match the field names in the table that you are appending to.
- The data in a field is inappropriate for the data type of the destination field. For example, the destination field has the Date/Time data type in an Access

- database— or the datetime data type in an Access project— but the data contains a text value that Microsoft Access can't recognize as a date or time.
- The data in a numeric field is too large for the field size of the destination field. For example, the destination field has the **FieldSize** property set to **Byte** in an Access database— or a data type of tinyint in an Access project — but the data contains a value greater than 255.
 - Records that you are importing contain duplicate values that cannot be stored in the [primary key](#) of the destination table or in any field in the table that has the **Indexed** property set to **Yes (No Duplicates)** in an Access database or that has the option **Ignore Duplicate Key** in an Access project.
 - One or more rows in the text file or spreadsheet contain more fields than the destination table.
 - The date fields that are imported from a Microsoft Excel spreadsheet are off by four years. Microsoft Excel for Windows uses the 1900 Date System, in which serial numbers range from 1 to 65,380, which correspond to the dates January 1, 1900 through December 31, 2078. However, Microsoft Excel for the Macintosh uses the 1904 Date System, in which serial numbers range from 0 to 63,918, which correspond to the dates January 1, 1904 through December 31, 2078.

Before importing the data, change the date system for the Microsoft Excel spreadsheet or, after appending the data, perform an update query using the expression `[date field name] + 1462` to correct the dates.

[I can't open or update a linked Microsoft Visual FoxPro or dBASE table.](#)

- If you link a .dbf file and associate an index (.idx, .cdx, .ndx, or .mdx) file, then Microsoft Access needs the index file to open the linked table. If you delete or move Microsoft Visual FoxPro or dBASE index files or the information (.inf) file Microsoft Access creates, you won't be able to open the linked table.
- If you link a Visual FoxPro or dBASE table with a memo field, then Microsoft Access needs the associated memo (.dbt) file to open the linked table. If you delete or move this file, you won't be able to open the linked table.
- When you use Microsoft Access to update the data in your .dbf file, it updates the index files to reflect your changes. If you use Visual FoxPro or dBASE to update data, you must update the associated indexes within Visual FoxPro or dBASE as well. Microsoft Access can't use a linked table

if the indexes you specified aren't current.

- You can't update a Visual FoxPro or dBASE table if its data (.dbf) or index (.idx, .cdx, .ndx, or .mdx) files are set to read-only. To check this, use Microsoft Windows Explorer to display the folder where the files are located, right-click the data or index file, click **Properties**, and then click the **General** tab. If the **Read-only** check box is selected, clear it and then click **OK**.
- If your Visual FoxPro or dBASE tables are stored on a read-only drive or CD-ROM, Microsoft Access can't create an .inf file in the same folder as the .dbf files, and you won't be able to link the tables. To link a table on a read-only drive, you must specify in the Windows Registry the path to a read/write drive where you want Microsoft Access to create the .inf file.
- You may not have the latest dBASE ISAM drivers. Microsoft has created updated dBASE ISAM drivers for Microsoft Jet 4.0 that do not require the installation of the Borland Database Engine (BDE) to provide full read/write access to dBASE files. The default Jet 4.0-based dBase ISAM drivers that ship with Microsoft Data Access Component (MDAC) 2.1 and later, allow read-only access to dBASE files unless the BDE is installed. To obtain these ISAM drivers, you must contact Microsoft Technical Support, and request that they be sent to you by a Support Professional.

[I can't open or update a linked Paradox table.](#)

- If you link a Paradox table that has a [primary key](#), then Microsoft Access needs the associated index (.px) file in order to open the linked table. If you link a Paradox table with a Memo field, then Microsoft Access needs the associated memo (.mb) file in order to open the linked table. If you delete or move these files, then you won't be able to open the linked table.
- If you link a Paradox table that doesn't have a primary key, then you won't be able to update data in the table using Microsoft Access. If you want the table to be updatable, define a primary key for the table in Paradox.
- [OLE objects](#) stored in a linked or [imported](#) Paradox table can't be opened from within Microsoft Access.
- You can't update a Paradox table if its data (.db) or index file (.px) is set to read-only. To check this, use Microsoft Windows Explorer to display the folder where the files are located, right-click the data or index file, click **Properties**, and then click the **General** tab. If the **Read-only** check box is selected, clear it and then click **OK**.
- You may not have the latest drivers. Microsoft has created updated Paradox

ISAM drivers for Microsoft Jet 4.0 that do not require the installation of the Borland Database Engine (BDE) to provide full read/write access to Paradox files. The default Jet 4.0-based Paradox ISAM drivers that ship with Microsoft Data Access Component (MDAC) 2.1 and later allow read-only access to Paradox files unless the BDE is installed. To obtain these ISAM drivers, you must contact Microsoft Technical Support, and request that they be sent to you by a Support Professional.

- When you link a Paradox table that resides on a server and is shared by multiple users, you must set the **ParadoxNetPath** key in the Windows Registry to the path to the Paradox.net or Pdoxusrs.net file. Additionally, if you use Paradox version 4.x for sharing data in your workgroup, you must set the **ParadoxNetStyle** key to "4.x."
- If you are using Microsoft Access to link Paradox data on a network while users of Paradox are using the data, you must be sure to define all three **ParadoxUserName**, **ParadoxNetPath**, and **ParadoxNetStyle** settings in the registry. Use the **ParadoxUserName** setting to specify a unique user name for accessing the shared Paradox data. Use the **ParadoxNetPath** setting to specify the full path to the folder containing the Paradox.net file (for Paradox 3.x) or the Pdoxusrs.net file (for Paradox 4.x and 5.0). The **ParadoxNetPath** setting (including the drive letter) must be consistent for all users sharing a particular database. Use the **ParadoxNetStyle** setting to specify the network locking style to use when accessing Paradox data. Possible values are "3.x" for Paradox 3.x and "4.x" for Paradox 4.x and "5.0" for Paradox 5.0.

 [I can't open or update a linked SQL table.](#)

- Microsoft Access requires a [unique index](#) to update, delete, or insert data in a linked SQL table or view. When you link a Microsoft SQL Server updatable view, or an [SQL database](#) table that doesn't have a unique index, Microsoft Access asks you to select one or more fields to generate a unique index. However, Microsoft Access doesn't test to make sure that the fields you selected uniquely identify each record. If there are duplicates in the index Microsoft Access builds, you won't be able to update those records.

If you think that the fields you selected did not create a unique index, delete the link to the table, link it again, and then select the appropriate field or fields to establish a unique index.

Alternatively, you can create an index within Microsoft Access by using the **Create Index** statement to create a [data-definition query](#). Again, be sure to select a field or combination of fields that will uniquely identify each record. If the index that is generated contains any duplicate values, all updates to the table will fail. To delete the index, use another data-definition query.

- If you link an SQL table that has its own unique index, and that index is defined for a field that is a floating-point number, the table might not be updatable. Because servers vary in how precisely they handle floating-point data, precision loss can sometimes occur. The actual difference is usually slight enough to be inconsequential, but if the data forms part of the SQL table's bookmark, affected records might appear to Microsoft Access to be deleted.
- If you encounter other errors while importing, linking, or using an SQL database table, there might be a problem with your account on the SQL database server or with the database itself. If you can't access an SQL database table, contact the SQL database administrator.

[I get a key violation message.](#)

A key violation occurs if:

- You attempt to run a query that appends, deletes, or updates records in a way that violates the rules of [referential integrity](#) for related tables.
- Your query attempts to append or update records that contain [primary key](#) values that already exist in the destination table.

If you run such a query, it will not modify the records that cause the key violation.

To modify records in a way that will violate referential integrity, you can break the relationship between the affected tables or turn referential integrity off, and then modify the affected tables one at a time. However, the resulting data in the two tables may then conflict.

[I imported a file and now I have an Import Errors table.](#)

The Import Errors table contains descriptions of most error messages Microsoft

Access encounters when trying to import your text file or spreadsheet in a Microsoft Access database. The table includes field names and row numbers that indicate which data has caused errors.

If Microsoft Access reports errors, open the Import Errors table and try to determine why Microsoft Access can't import all of the records. The following table lists possible import errors and describes their causes.

Error	Description
Field Truncation	A value in the file is too large for the FieldSize property setting for this field.
Type Conversion Failure	A value in the text file or spreadsheet is the wrong data type for this field.
Key Violation	This record's primary key value is a duplicate— it already exists in the table.
Validation Rule Failure	A value breaks the rule set by using the ValidationRule property for this field or for the table.
Null in Required Field	A Null value isn't allowed in this field because the Required property for the field is set to Yes .
Null value in AutoNumber field	The data that you're importing contains a Null value that you attempted to append to an AutoNumber field.
Unparsable Record	A text value contains the text delimiter character (usually double quotation marks). Whenever a value contains the delimiter character, the character must be repeated twice in the text file; for example: 10 - 3 1/2"" disks/box

If you suspect that the problem is with your data, edit your text file or spreadsheet. If you're appending records to an existing table, you might need to change the table itself (reorder the fields or change property settings, for example). When you have solved the problem, import the data again.

[I'm trying to import or link data from my security-enabled database.](#)

If the database that you want to import or link has a database password, then you must provide the password before you can continue. Linking tables from such a database may have unforeseen consequences.

[File extensions don't show up beside the file names in the Files Of Type dialog box.](#)

You have not selected the **Show hidden files and folders** option in Windows Explorer. See Windows Help for information on selecting the option.

[I get an "ODBC – connection to *datasourcename* failed" message when I open a table or form in a converted database.](#)

The message "ODBC – connection to *datasourcename* failed" occurs when a table in your converted Microsoft Access version 1.x or 2.0 database is linked to an [ODBC data source](#) that uses a 16-bit driver manager and driver. Microsoft Access 2002 or later can open only linked ODBC data sources that use the 32-bit versions of the ODBC Driver Manager (Odbc32.dll) and the appropriate ODBC driver—for example, the 32-bit version of Microsoft SQL Server ODBC Driver (Sqlsrv32.dll).

To correct this problem, you must create a new, identically-named 32-bit data source name (DSN) for each ODBC data source that is linked to the original Access 1.x or 2.0 database.

1. In Microsoft Windows Control Panel, double-click **Administrative Tools**, and then double-click **Data Sources (ODBC)**.
2. Create the new DSN by entering values identical to the original DSN. For example, for Microsoft SQL Server, you need to define at least **Data Source Name** and **Server**, but you may also need to define additional values such as the **Database Name**.

If you are uncertain what the original name of the DSN was, open the original database in the version of Microsoft Access that it was created with, open the linked table in [Design view](#), and then display the **Table Properties** sheet. The table's **Description** property contains the definition for the ODBC connection string. The parameter following **DSN=** in the connection string is the name of the DSN.

[Show All](#)

About ODBC data sources

An [ODBC data source](#)— for example, a database and the server on which it resides— is accessed through an [Open Database Connectivity](#) (ODBC) driver.



[What is a data source?](#)

A data source is a source of data and the connection information needed to access that data. Examples of data sources are Microsoft Access, Microsoft SQL Server, Oracle RDBMS, a spreadsheet, and a text file. Examples of connection information include server location, database name, logon ID, password, and various ODBC driver options that describe how to connect to the data source.

In the ODBC architecture, an application (such as Access or a Microsoft Visual Basic program) connects to the ODBC Driver Manager, which in turn uses a specific ODBC driver (for example, Microsoft SQL ODBC driver) to connect to a data source (in this case, a [Microsoft SQL Server database](#)). In Access, you use ODBC data sources to connect to data sources external to Access that do not have built-in drivers.

To connect to these data sources, you must do the following:

- Install the appropriate ODBC driver on the computer that contains the data source.
- Define a data source name (DSN) by using either the **ODBC Data Source Administrator** to store the connection information in the Microsoft Windows registry or a DSN file, or a connect string in Visual Basic code to pass the connection information directly to the ODBC Driver Manager.



[Machine data sources](#)

Machine data sources store connection information in the Windows Registry on a specific computer with a user-defined name. You can use machine data sources only on the computer they are defined on. There are two types of machine data sources— user and system. User data sources can be used only by the current user and are visible only to that user. System data sources can be used by all

users on a computer and are visible to all users on the computer and system-wide services, such as Microsoft Windows services. A machine data source is especially useful when you want to provide added security, because it helps ensure that only users who are logged on can view a machine data source and it cannot be copied by a remote user to another computer.

[File data sources](#)

File data sources (also called DSN files) store connection information in a text file, not the Windows registry, and are generally more flexible to use than machine data sources. For example you can copy a file data source to any computer with the correct ODBC driver so that your application can rely on consistent and accurate connection information to all the computers it uses. Or you can place the file data source on a single server, share it between many computers on the network, and easily maintain the connection information in one location.

A file data source can also be unshareable. An unshareable file data source resides on a single computer and points to a machine data source. You can use unshareable file data sources to access existing machine data sources from file data sources.

[Connect strings](#)

If you are a programmer, you can define a formatted connect string in your Microsoft Visual Basic code that specifies the connection information. Using a connect string avoids defining a machine or file DSN and passes the connection information directly to the ODBC Driver Manager. This is useful, for example, when you want to avoid requiring system administrators or users to first create a DSN, or to simplify the installation of your application. To maintain security for the connection string information in your code, help protect your code by using a MDE file or protecting your code with a password.

Security Note Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

Set up or change ODBC data sources

1. Open Microsoft Windows Control Panel and then double-click the **Administrative Tools** icon. Double-click on the **Data Sources (ODBC)** icon.
2. Select the **User DSN**, **System DSN**, or **File DSN** tab, depending on the type of data source to be modified.
3. Do one of the following:

To define a new data source for a currently installed driver, click **Add**.

To modify the definition of an existing data source, select the data source from the list. If you are modifying a file DSN, find the appropriate folder in the **Look In** box (use the **Up** control if necessary), and then select the data source from the folder. Click **Configure**.

4. Complete the dialog boxes. For more information about the different options, click the **Help** button in each dialog box.

[Show All](#)

Differences between ODBC drivers and built-in drivers for external data

In Microsoft Access, you can [import](#), [export](#), or link data from a number of different database, spreadsheet, and text file formats. To do this, Access uses either a built-in driver or an [ODBC](#) driver.

Built-in drivers

Access contains built-in drivers that enable you to import, export, or link the following types of data: other Microsoft Access databases, Microsoft Access projects, Paradox, dBASE, Microsoft Excel, Microsoft Exchange, HTML, Lotus 1-2-3 spreadsheets, Microsoft Outlook, Microsoft Windows SharePoint Services, fixed-width text files, delimited text files, and XML.

If a built-in driver for a type of data has been installed, that type will be available in the **Export**, **Import** or **Link** dialog boxes. All built-in drivers are installed automatically when you install Access. ODBC drivers may also be installed on your computer for the products listed above, but Microsoft recommends you use a corresponding built-in driver when available.

ODBC drivers

You can use ODBC drivers to connect to Microsoft SQL Server databases, Microsoft FoxPro databases, and data from other programs that provide 32-bit drivers in compliance with ODBC Level 1 to access their data files. Microsoft Access automatically installs the Microsoft SQL Server ODBC driver (Sqlsrv32.dll), the Microsoft FoxPro ODBC driver (vfpoDBC.dll), the Oracle ODBC driver (msorcl32.dll), and some ODBC support files (Help files and the Control Panel/Administrative Tools **Data Sources (ODBC)** icon).

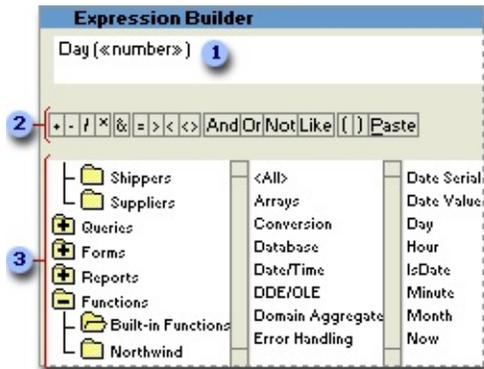
Third-party vendors may supply additional ODBC drivers and documentation that support other data formats. For more information about whether an ODBC driver is available for a particular type of data, contact the vendor of the program that produces that type of data. Access requires 32-bit ODBC drivers that are in compliance with ODBC Level 1.

Other Microsoft products and products from other vendors may also contain ODBC drivers. If you want to know whether these ODBC drivers have been tested and verified for use with Access, contact the driver vendor.

[Show All](#)

About the Expression Builder

The Expression Builder has three sections, located from top to bottom:



1 Expression box In the upper section of the builder is an expression box where you build the [expression](#). Use the lower section of the builder to create elements of the expression, and then paste these elements into the expression box to form an expression. You can also type parts of the expression directly into the expression box.

2 Operator buttons In the middle section of the builder are buttons for commonly used [operators](#). If you click one of the operator buttons, the Expression Builder inserts the operator at the insertion point in the expression box. For a complete list of operators you can use in expressions, click the **Operators** folder in the lower-left box and the appropriate operator category in the middle box. The right box lists all operators in the selected category.

3 Expression elements In the lower section of the builder are three boxes:

- The left box contains folders that list the table, query, form, and report database objects, built-in and user-defined functions, [constants](#), operators, and common expressions.
- The middle box lists specific elements or categories of elements for the folder selected in the left box. For example, if you click **Built-In Functions** in the left box, the middle box lists categories of Microsoft Access functions.
- The right box lists the values, if any, for the elements you select in the left

and middle boxes. For example, if you click **Built-In Functions** in the left box and a category of functions in the middle box, the right box lists all built-in functions in the selected category.

Note When you paste an [identifier](#) in your expression, the Expression Builder pastes only the parts of the identifier that are required in the current context. For example, if you start the Expression Builder from the [property sheet](#) of the Customers form, and then paste an identifier for the **Visible** property of the form in your expression, the Expression Builder pastes only the property name: **Visible**. If you use this expression outside the context of the form, you must include the full identifier: Forms![Customers].Visible

[Show All](#)

About expressions

[Expressions](#) are a fundamental part of many Microsoft Access operations. An expression is a combination of symbols— [identifiers](#), [operators](#), and values — that produce a result. For example, you can use the following expression in a [control](#) on a form or report to display the sum of the values in the Subtotal and Freight controls:

= [Subtotal] + [Freight]



[When to use expressions](#)

- To set a property that defines a [calculated control](#), establishes a [validation rule](#), or that sets a default field value.
- To enter a [criteria](#) expression, create a [calculated field](#), or update records in a query or filter.
- To set a condition for carrying out an [action](#) or series of actions in a macro, or to specify [arguments](#) for many actions.
- To specify arguments for many functions, [statements](#), and [methods](#) in [Microsoft Visual Basic for Applications](#) procedures.
- To edit an [SQL](#) query in [SQL view](#) of the [Query window](#), or to use an [SQL statement](#) in a property setting or argument.

You can specify a value in an expression by using a literal value, a constant, a function, or an identifier.

[Literal values](#)

A literal value represents a value— such as a number, string, or date— that Microsoft Access evaluates exactly as written. "New York," 100, and #1-Jan-01# ('1-Jan-01' in ANSI-92) are examples of literal values.

[Date/time values](#)

Number signs (#) (or single (') quotation marks in ANSI-92) around an element of an expression indicate that the element is a date/time value. Microsoft Access automatically evaluates a value surrounded by number signs (or single quotation marks) as a date/time value and lets you type the value in any common date or time format.

You don't have to type number signs (or single quotation marks) around a date/time value in a validation expression or a criteria expression for a field whose [data type](#) is Date/Time. You can type the value in any common date or time format and Microsoft Access automatically inserts the correct signs around the value. Note that Access 2002 and later conforms to ANSI-92.

Microsoft Access displays the value according to the regional settings in Microsoft Windows Control Panel. You can change the output format of a date by using the **Format** property.

[Text strings](#)

Double (") quotation marks around an element of an expression indicate that the element is text.

When you type text in a [validation](#) or [criteria](#) expression, you can type the text without double quotation marks and Microsoft Access will insert them automatically.

For example, if you type the expression **Paris**, Microsoft Access displays the

expression as follows:

"Paris"

If you want an expression to produce a string that is enclosed in double quotation marks, you can either enclose the nested string in single (') quotation marks or three sets of double (") quotation marks. For example, the following expressions are equivalent:

```
Forms![Contacts]![City].DefaultValue = ' "Paris" '
```

```
Forms![Contacts]![City].DefaultValue = " " "Paris" " "
```

[Constants](#)

A constant represents a value that doesn't change. **True**, **False**, and **Null** are examples of constants that are defined automatically by Microsoft Access. You can also define your own constants in [Microsoft Visual Basic for Applications](#) that you can use in Visual Basic procedures.

Note You cannot use Visual Basic constants in functions used in your expressions. For example, Visual Basic has constants for the days of the week that can be used in a function such as **vbSunday** to represent Sunday, **vbMonday** to represent Monday, and so forth. Each constant has a corresponding numeric value: the numeric value for **vbSunday** is 1 and the numeric value for **vbMonday** is 2. In a function used in an expression, you must use the numeric values to represent the particular day of the week.

[Functions](#)

A function returns a value based on the results of a calculation or other operation. Microsoft Access includes many built-in functions; for example:

- The **Date** function returns the current date.
- The **Sum** function returns the sum of a set of field values.
- The **DLookup** function returns a specific field value.

[Field, control, and property identifiers](#)

An identifier refers to the value of a field, control, or property. For example, the following identifier refers to the value of the **DefaultValue** property of the OrderDate control on the Orders form:

```
Forms![Orders]![OrderDate].DefaultValue
```

About building expressions

[Entering object names](#)

Brackets ([]) around a field, control, or property in an identifier indicate that the element is the name of a table, query, form, report, field, or control.

When you type an object name in an identifier, you surround the name with brackets if it contains a space or a special character, such as an underscore. If the name doesn't contain a space or a special character, you can type the name without brackets. Microsoft Access inserts brackets automatically (with two exceptions noted below).

For example, you can type the following expression as a setting for the **ControlSource** property to calculate the sum of the values in the Freight and OrderAmount fields:

= Freight + OrderAmount

Microsoft Access displays the expression as follows:

= [Freight] + [OrderAmount]

Note In a **ValidationRule** property setting or in a **Criteria** cell in the query [design grid](#), Microsoft Access doesn't always automatically insert brackets around a name. If you are entering an object name, be sure to type brackets around it. Otherwise, Microsoft Access may assume you are entering text and may insert double quotation marks.

[Using the ! and . \(dot\) operators in expressions](#)

You use the ! and . (dot) operators in an identifier to indicate the type of item that immediately follows.

The ! operator

The ! operator indicates that what follows is a user-defined item (an element of a [collection](#)). For example, use the ! operator to refer to an open form, an open report, or a control on an open form or report.

Identifier	Refers to
Forms![Orders]	The open Orders form.
Reports![Invoice]	The open Invoice report.
Forms![Orders]![OrderID]	The OrderID control on the open Orders form.

The . (dot) operator

The . (dot) operator usually indicates that what follows is an item defined by Microsoft Access. For example, use the . (dot) operator to refer to a property of a form, report, or control. You can also use the . (dot) operator to refer to a field value in an [SQL statement](#), a Microsoft Visual Basic for Applications [method](#), or a collection.

Identifier	Refers to
Reports![Invoice]![ShipName].Visible	The Visible property of the ShipName control on the Invoice report.
SELECT Employees.EmployeeID, Orders.OrderID FROM Employees INNER JOIN Orders ON Employees.EmployeeID = Orders.EmployeeID;	The EmployeeID field from the Employees table and the Orders table.
DoCmd.Close	The Close method in Microsoft Visual Basic .
Forms![Orders].Properties.Refresh	The Refresh method of the Properties collection of the Orders form.

[Combining values of fields, controls, or properties](#)

You can combine the value of a field, control, or property with a literal string by using the **&** (concatenation) operator. For example, the following expression combines the literal string "[CategoryID] = " with the value of the CategoryID control on the Products form:

```
"[CategoryID] = " & Forms![Products]![CategoryID]
```

In some circumstances— for example, in a [domain aggregate function](#) such as **DLookup**— the value of the field, control, or property must appear in single (') or double (") quotation marks. The easiest way to accomplish this is to add a single quotation mark to the literal string, and then combine the expression with

another literal string made up of a single quotation mark after the field, control, or property value, as follows:

```
"[CategoryID] = ' " & Forms![Products]![CategoryID] & " ' "
```

Existing Microsoft Access applications may use the vertical bar operators (| |) in place of an opening and closing combination of double quotation marks and & (concatenation) operators, as follows:

```
"[CategoryID] = '|Forms![Products]![CategoryID]||' "
```

However, the use of vertical bars is not recommended because they can produce unexpected results in some circumstances.

You can either create the expression on your own or use the Expression Builder.

 [About creating an expression on your own](#)

If you are entering an expression that is longer than the standard input area in a [property sheet](#), a [design grid](#), or an [action argument](#), you can type the expression in the **Zoom** box. To open the **Zoom** box, press SHIFT+F2 when the [focus](#) is where you want to enter the expression.

When you enter an expression in a property sheet, a design grid, or an action argument, Microsoft Access:

- Inserts certain characters for you when the focus changes. Depending on where you enter the expression, Microsoft Access automatically inserts brackets ([]) around form, report, field, or [control](#) names, number signs (#) around dates, and double (") quotation marks around text.

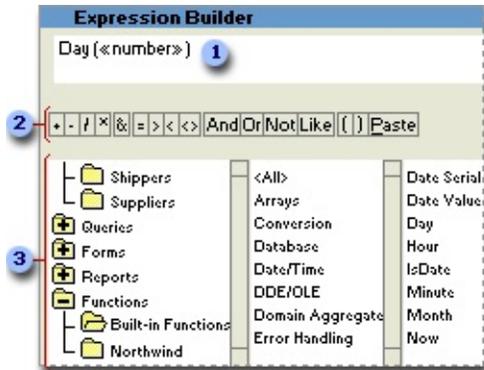
Note When you add an expression to a [calculated control](#), you must add an equal sign (=) before the expression. Access does not insert the equal sign for you.

- Recognizes region-specific function names, property names, and list separators for international versions. When you enter a function or property in an expression in most international versions of Microsoft Access, you can:
 - Type its localized name (in your language) in a property sheet, a design grid, or an action argument.
 - Use the list separator for your country/region when you specify more than one [argument](#) for a function. You specify the list separator on the **Number** tab as part of the regional settings in Microsoft Windows Control Panel. For most international versions, the default list separator is a semicolon (;).

However, in Microsoft Visual Basic for Applications code, you must type the English function or property name and use a comma (,) as a list separator.

About the Expression Builder

The Expression Builder has three sections, located from top to bottom:



1 Expression box In the upper section of the builder is an expression box where you build the [expression](#). Use the lower section of the builder to create elements of the expression, and then paste these elements into the expression box to form an expression. You can also type parts of the expression directly into the expression box.

2 Operator buttons In the middle section of the builder are buttons for commonly used [operators](#). If you click one of the operator buttons, the Expression Builder inserts the operator at the insertion point in the expression box. For a complete list of operators you can use in expressions, click the **Operators** folder in the lower-left box and the appropriate operator category in the middle box. The right box lists all operators in the selected category.

3 Expression elements In the lower section of the builder are three boxes:

- The left box contains folders that list the table, query, form, and report database objects, built-in and user-defined functions, [constants](#), operators, and common expressions.
- The middle box lists specific elements or categories of elements for the folder selected in the left box. For example, if you click **Built-In Functions** in the left box, the middle box lists categories of Microsoft Access functions.
- The right box lists the values, if any, for the elements you select in the left and middle boxes. For example, if you click **Built-In Functions** in the left box and a category of functions in the middle box, the right box lists all built-in functions in the selected category.

Note When you paste an [identifier](#) in your expression, the Expression Builder pastes only the parts of the identifier that are required in the current context. For example, if you start the Expression Builder from the [property sheet](#) of the Customers form, and then paste an identifier for the **Visible** property of the form in your expression, the Expression Builder pastes only the property name: **Visible**. If you use this expression outside the context of the form, you must include the full identifier: Forms![Customers].Visible

[Show All](#)

About Microsoft Jet Expression Service sandbox mode

Microsoft Office Access 2003 uses the Microsoft Jet Expression Service to evaluate [expressions](#). The Jet Expression Service can be configured to run in sandbox mode. Sandbox mode is a mode of operation where only safe expressions can be evaluated.

About unsafe expressions

Unsafe expressions contain functions that could be exploited by malicious users to access drives, files, or other resources for which they do not have authorization. If malicious users have access to these resources, then they may be able to perform actions such as deleting all of the files from a folder, tying up a network resource, or modifying a file.

To make your Access databases (MDBs) and projects (ADPs) more resistant to malicious attacks, you should enable sandbox mode. Sandbox mode allows Access to be fully functional while blocking unsafe expressions.

Functions that are blocked in sandbox mode

The following functions are blocked when the Jet Expression Service is configured to run in sandbox mode. the functions cannot be used in expressions, but they can be utilized in Visual Basic for Applications (VBA) code.

 [Blocked functions located in the Visual Basic for Applications \(VBA\) object library](#)

Function Name

AppActivate
Beep
Calendar
CallByName
ChDir
ChDrive
Command
Command\$
CreateObject
CurDir
CurDir\$
DeleteSetting
DoEvents
Environ
Environ\$
EOF
Err
FileAttr
FileCopy
FileDateTime
FileLen
FreeFile
GetAllSettings

GetAttr
GetObject
GetSetting
Input
Input\$
InputB
InputB\$
Kill
Load
Loc
LOF
Randomize
Reset
SaveSetting
Seek
SendKeys
SetAttr
Shell
Spc
Tab
Unload
UserForms
Width

[Blocked functions and properties in the Microsoft Access object library](#)

Object	Blocked Function/Property
Application	AddAutoCorrect
Application	AddToFavorites
Application	ADODConnectString
Application	AnswerWizard
Application	Application
Application	Assistant

Application	AutoCorrect
Application	BeginUndoable
Application	CloseCurrentDatabase
Application	CodeContextObject
Application	CodeDb
Application	COMAddIns
Application	CommandBars
Application	CompactRepair
Application	ConvertAccessProject
Application	CreateAccessProject
Application	CreateAdditionalData
Application	CreateControl
Application	CreateControlEx
Application	CreateDataAccessPage
Application	CreateForm
Application	CreateGroupLevel
Application	CreateNewWorkgroupFile
Application	CreateReport
Application	CreateReportControl
Application	CreateReportControlEx
Application	CurrentDb
Application	DataAccessPages
Application	DBEngine
Application	DDEExecute
Application	DDEInitiate
Application	DDEPoke
Application	DDERequest
Application	DDETerminate
Application	DDETerminateAll
Application	DefaultWebOptions
Application	DefaultWorkspaceClone
Application	DelAutoCorrect
Application	DeleteControl

Application	DeleteReportControl
Application	DoCmd
Application	Echo
Application	ExportXML
Application	FeatureInstall
Application	FileDialog
Application	FileSearch
Application	FollowHyperlink
Application	GetHiddenAttribute
Application	ImportXML
Application	InsertText
Application	LanguageSettings
Application	LoadFromText
Application	LoadPicture
Application	Modules
Application	NewAccessProject
Application	NewCurrentDatabase
Application	NewFileTaskPane
Application	OpenAccessProject
Application	OpenCurrentDatabase
Application	Parent
Application	ProductCode
Application	Quit
Application	References
Application	RefreshDatabaseWindow
Application	RefreshTitleBar
Application	ReloadAddIns
Application	ReplaceModule
Application	RunCommand
Application	SaveAsText
Application	SetDefaultWorkgroupFile
Application	SetHiddenAttribute
Application	SetOption

Application	SetUndoRecording
Application	SysCmd
Application	TransformXML
Application	VBE
BoundObjectFrame	Object
Combobox	Recordset
Control	Object
CurrentProject	AccessConnection
CurrentProject	BaseConnectionString
CurrentProject	CloseConnection
CurrentProject	Connection
CurrentProject	OpenConnection
CustomControl	Object
Form	Dynaset
Form	Recordset
Form	RecordsetClone
Hyperlink	AddToFavorites
Hyperlink	CreateNewDocument
Hyperlink	Follow
ListBox	Recordset
ObjectFrame	Object
Report	Recordset
SmartTagAction	Execute
Screen	ActiveDataAccessPage

How to enable sandbox mode

Follow these steps to enable sandbox mode:

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Security Level** tab, and then click **Medium** or **High**.
3. Click **OK**.
4. Restart Access.

Note Enabling the sandbox mode setting applies to all users of the computer.

When are unsafe expressions blocked?

These functions are blocked only when used in expressions such as for a default value, a control source, or in a SQL statement. These functions are not blocked for use in VBA code.

How to use a blocked function in an expression

You can utilize a blocked function in an expression by wrapping the function in a public VBA function. For example, you can create the following VBA function in order to use the blocked CurDir function:

```
Public Function MyCurDir() As String
    MyCurDir = CurDir()
End Function
```

You would call the MyCurDir() function in your expression instead of calling the CurDir() function.

How to disable sandbox mode

Follow these steps to disable sandbox mode:

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Security Level** tab, and then click **Low**.
3. Click **OK**.
4. Restart Access.

Note At this security level, macros are always enabled when you open files.

Note Disabling the sandbox mode setting applies to all users of the computer.

[Show All](#)

Functions that are blocked in sandbox mode

The following functions are blocked when Jet is configured to run in sandbox mode. the functions cannot be used in [expressions](#), but they can be utilized in Visual Basic for Applications (VBA) code.

 [Blocked functions located in the Visual Basic for Applications \(VBA\) object library](#)

Function Name

AppActivate

Beep

Calendar

CallByName

ChDir

ChDrive

Command

Command\$

CreateObject

CurDir

CurDir\$

DeleteSetting

DoEvents

Environ

Environ\$

EOF

Err

FileAttr

FileCopy

FileDateTime

FileLen
FreeFile
GetAllSettings
GetAttr
GetObject
GetSetting
Input
Input\$
InputB
InputB\$
Kill
Load
Loc
LOF
Randomize
Reset
SaveSetting
Seek
SendKeys
SetAttr
Shell
Spc
Tab
Unload
UserForms
Width

 [Blocked functions and properties in the Microsoft Access object library](#)

Object	Blocked Function/Property
Application	AddAutoCorrect
Application	AddToFavorites
Application	ADODefaultConnectionString

Application	AnswerWizard
Application	Application
Application	Assistant
Application	AutoCorrect
Application	BeginUndoable
Application	CloseCurrentDatabase
Application	CodeContextObject
Application	CodeDb
Application	COMAddIns
Application	CommandBars
Application	CompactRepair
Application	ConvertAccessProject
Application	CreateAccessProject
Application	CreateAdditionalData
Application	CreateControl
Application	CreateControlEx
Application	CreateDataAccessPage
Application	CreateForm
Application	CreateGroupLevel
Application	CreateNewWorkgroupFile
Application	CreateReport
Application	CreateReportControl
Application	CreateReportControlEx
Application	CurrentDb
Application	DataAccessPages
Application	DBEngine
Application	DDEExecute
Application	DDEInitiate
Application	DDEPoke
Application	DDERequest
Application	DDETerminate
Application	DDETerminateAll
Application	DefaultWebOptions

Application	DefaultWorkspaceClone
Application	DelAutoCorrect
Application	DeleteControl
Application	DeleteReportControl
Application	DoCmd
Application	Echo
Application	ExportXML
Application	FeatureInstall
Application	FileDialog
Application	FileSearch
Application	FollowHyperlink
Application	GetHiddenAttribute
Application	ImportXML
Application	InsertText
Application	LanguageSettings
Application	LoadFromText
Application	LoadPicture
Application	Modules
Application	NewAccessProject
Application	NewCurrentDatabase
Application	NewFileTaskPane
Application	OpenAccessProject
Application	OpenCurrentDatabase
Application	Parent
Application	ProductCode
Application	Quit
Application	References
Application	RefreshDatabaseWindow
Application	RefreshTitleBar
Application	ReloadAddIns
Application	ReplaceModule
Application	RunCommand
Application	SaveAsText

Application	SetDefaultWorkgroupFile
Application	SetHiddenAttribute
Application	SetOption
Application	SetUndoRecording
Application	SysCmd
Application	TransformXML
Application	VBE
BoundObjectFrame	Object
Combobox	Recordset
Control	Object
CurrentProject	AccessConnection
CurrentProject	BaseConnectionString
CurrentProject	CloseConnection
CurrentProject	Connection
CurrentProject	OpenConnection
CustomControl	Object
Form	Dynaset
Form	Recordset
Form	RecordsetClone
Hyperlink	AddToFavorites
Hyperlink	CreateNewDocument
Hyperlink	Follow
ListBox	Recordset
ObjectFrame	Object
Report	Recordset
SmartTagAction	Execute
Screen	ActiveDataAccessPage

[Show All](#)

Troubleshoot expressions

Unexpected results

Although using [expressions](#) is relatively straightforward, you can occasionally get unexpected results when you:

[Combine \(concatenate\) control or field values, variables, or text in an expression.](#)

Concatenating [control](#) or field values or text in an expression might produce unexpected results for any of the following reasons:

- You're referring to a control or field that has the same name as the current control, creating a circular reference. Change the name of the field or control.
- You're referring to a control or field using a name that is misspelled or typed incorrectly. Verify that it's entered correctly.
- You're referring to a control or field that has the same name as a property. Use the full [identifier](#) for the control or field.
- Microsoft Access isn't evaluating the expression correctly because part of the expression must be surrounded by brackets ([]) or single (') or double (") quotation marks. Check to see whether bracket or quotation mark characters are required.

[Perform operations on fields that have blank values.](#)

Many operations might produce unexpected results if they use an expression that refers to a field that has blank (**Null**) values. For example, the **DCount** or **DSum** functions might produce unexpected results if they use [criteria](#) expressions that refer to a field with **Null** values.

To correct this problem, you can either:

- Pass field values to the **Nz** function in the expression. When the **Nz** function evaluates a **Null** value, the function returns zero or a [zero-length](#)

[string](#).

- Use the **IIf** and **IsNull** functions to test whether a value is **Null**, and then return an appropriate value.

Enter a date value in an expression.

Microsoft Access can produce unexpected results if you enter a date value in an expression without surrounding it with number signs (#) for ANSI-89 or single (') quotation marks in ANSI-92. Note that Access 2002 and later conforms to ANSI-92.



[Refer to a subform or subreport property or control](#).

Microsoft Access can produce unexpected results if you refer to a subform or subreport property or [control](#) by using an incorrect [identifier](#).

To refer to a subform or subreport property, you must type a full identifier for the form or report by using the **Form** or **Report** property of a subform or subreport control. For example, the following identifier refers to the **Visible** property of the Orders Subform subform:

Forms![Orders]![OrdersSubformControl].Form.Visible

Expression	Refers to
Forms![Orders]! [OrdersSubformControl]	The subform control that contains the subform.
Forms![Orders]! [OrdersSubformControl]	The subform itself. You must specify the Form property to refer to the subform itself or one of its properties.
.Form	

In contrast, to refer to a control on a subform or subreport, or one of the control's properties, you don't have to use the **Form** or **Report** property. For example, the following identifier refers to the **Visible** property of the Discount control on the Orders Subform subform:

Forms![Orders]![OrdersSubformControl]![Discount].Visible

[Refer to a column in a list box or combo box](#)

To refer to a column other than the bound column, use the **Column** property. The **Column** property is zero-based. For example, to refer to the second column in the combo box called SuppliersCombo on the Products form, use this syntax:

```
Forms![Products]![SuppliersCombo].Column(1)
```

Expression Builder

[I can't find a field name in the Expression Builder.](#)

To view new field names in the Expression Builder, you must first save the table or query containing the new fields.

[I can't find the object or function I want in the Expression Builder.](#)

If an object or function isn't listed in the lower section of the Expression Builder, it's not valid in the context where you started the builder. For example, you can't refer to other fields and controls in the [validation rule](#) for a field in table [Design view](#), so the **Tables**, **Queries**, **Forms**, and **Reports** folders aren't available if you start the Expression Builder from the **ValidationRule** property of a field in table Design view.

Glossary

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [N](#)
[O](#) [P](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [Z](#)

A

[Back to top](#)

access key

A key pressed while holding down the ALT key that allows the user to open a menu, carry out a command, select an object, or move to an object. For example, ALT+F opens the **File** menu.

ActiveX control

An object that you place on a form to enable or enhance a user's interaction with an application. ActiveX Controls have events and can be incorporated into other controls. These controls have an .ocx file name extension.

ActiveX object

An object that is exposed to other applications or programming tools through Automation interfaces.

add-in

A customized tool that adds capabilities to the Visual Basic development environment.

ANSI Character Set

American National Standards Institute (ANSI) 8-bit character set used to represent up to 256 characters (0 - 255) using your keyboard. The first 128 characters (0 - 127) correspond to the letters and symbols on a standard U.S. keyboard. The second 128 characters (128 - 255) represent special characters, such as letters in international alphabets, accents, currency symbols, and fractions.

argument

A constant, variable, or expression passed to a procedure.

array

A set of sequentially indexed elements having the same intrinsic data type. Each element of an array has a unique identifying index number. Changes made to one element of an array don't affect the other elements.

ASCII Character Set

American Standard Code for Information Interchange (ASCII) 7-bit character set used to represent letters and symbols found on a standard U.S. keyboard. The ASCII character set is the same as the first 128 characters (0 - 127) in the ANSI character set.

automatic formatting

A feature that automatically formats code as you enter it by capitalizing the first letter for keywords, standardizing spacing, adding punctuation, and setting the foreground and background colors.

Automation object

An object that is exposed to other applications or programming tools through Automation interfaces.

application

A collection of code and visual elements that work together as a single program. Developers can build and run applications within the development environment, while users usually run applications as executable files outside the development environment.

B

[Back to top](#)

base class

Original class from which other classes can be derived by inheritance.

bitmap

An image represented by pixels and stored as a collection of bits in which each bit corresponds to one pixel. On color systems, more than one bit corresponds to each pixel. A bitmap usually has a .bmp file name extension.

bitwise comparison

A bit-by-bit comparison between identically positioned bits in two numeric expressions.

Boolean expression

An expression that evaluates to either **True** or **False**.

Boolean data type

A data type with only two possible values, **True** (-1) or **False** (0). **Boolean** variables are stored as 16-bit (2-byte) numbers.

bound control

A data-aware control that can provide access to a specific field or fields in a database through a **Data** control. A data-aware control is typically bound to

a **Data** control through its **DataSource** and **DataField** properties. When a **Data** control moves from one record to the next, all bound controls connected to the **Data** control change to display data from fields in the current record. When users change data in a bound control and then move to a different record, the changes are automatically saved in the database.

breakpoint

A selected program line at which execution automatically stops.

Breakpoints are not saved with your code.

by reference

A way of passing the address of an argument to a procedure instead of passing the value. This allows the procedure to access the actual variable. As a result, the variable's actual value can be changed by the procedure to which it is passed. Unless otherwise specified, arguments are passed by reference.

Byte data type

A data type used to hold positive integer numbers ranging from 0 - 255.

Byte variables are stored as single, unsigned 8-bit (1-byte) numbers.

by value

A way of passing the value of an argument to a procedure instead of passing the address. This allows the procedure to access a copy of the variable. As a result, the variable's actual value can't be changed by the procedure to which it is passed.

break mode

Temporary suspension of program execution in the development environment. In break mode, you can examine, debug, reset, step through, or continue program execution. You enter break mode when you: Encounter a breakpoint during program execution. Press CTRL+BREAK during program execution. Encounter a **Stop** statement or untrapped run-time error during program execution. Add a **Break When True** watch expression. Execution stops when the value of the watch changes and evaluates to **True**. Add a **Break When Changed** watch expression. Execution stops when the value of the watch changes.

C

[Back to top](#)

character code

A number that represents a particular character in a set, such as the ANSI

character set.

class

The formal definition of an object. The class acts as the template from which an instance of an object is created at run time. The class defines the properties of the object and the methods used to control the object's behavior.

class module

A module that contains the definition of a class, including its property and method definitions.

code module

A module containing public code that can be shared among all modules in a project. A code module is referred to as a standard module in later versions of Visual Basic.

code pane

A pane contained in a code window that is used for entering and editing code. A code window can contain one or more code panes.

collection

An object that contains a set of related objects. An object's position in the collection can change whenever a change occurs in the collection; therefore, the position of any specific object in the collection can vary.

command line

The path, file name, and argument information provided by the user to run a program.

comment

Text added to code that explains how the code works. In Visual Basic, a comment line can start with either an apostrophe (') or with the **Rem** keyword followed by a space.

comparison operator

A character or symbol indicating a relationship between two or more values or expressions. These operators include less than (<), less than or equal to (<=), greater than (>), greater than or equal to (>=), not equal (<>), and equal (=). Additional comparison operators include **Is** and **Like**. Note that **Is** and **Like** can't be used as comparison operators in a **Select Case** statement.

compiler directive

A command used to alter the action of the compiler.

compile time

The period during which source code is translated to executable code.

conditional compiler constant

A Visual Basic identifier that is defined using the **#Const** compiler directive or defined in the host application and used by other compiler directives to determine when or if certain blocks of Visual Basic code are compiled.

constant

A named item that retains a constant value throughout the execution of a program. A constant can be a string or numeric literal, another constant, or any combination that includes arithmetic or logical operators except **Is** and exponentiation. Each host application can define its own set of constants. Additional constants can be defined by the user with the **Const** statement. You can use constants anywhere in your code in place of actual values.

container

An object that can contain other objects.

control

An object you can place on a form that has its own set of recognized properties, methods, and events. You use controls to receive user input, display output, and trigger event procedures. You can manipulate most controls using methods. Some controls are interactive (responsive to user actions), while others are static (accessible only through code).

control array

A group of controls that share a common name, type, and event procedures. Each control in an array has a unique index number that can be used to determine which control recognizes an event.

Currency data type

A data type with a range of -922,337,203,685,477.5808 to 922,337,203,685,477.5807. Use this data type for calculations involving money and for fixed-point calculations where accuracy is particularly important. The at sign (**@**) type-declaration character represents **Currency** in Visual Basic.

D

[Back to top](#)

data type

The characteristic of a variable that determines what kind of data it can hold. Data types include **Byte**, **Boolean**, **Integer**, **Long**, **Currency**, **Decimal**, **Single**, **Double**, **Date**, **String**, **Object**, **Variant** (default), and user-defined types, as well as specific types of objects.

Date data type

A data type used to store dates and times as a real number. Date variables are stored as 64-bit (8-byte) numbers. The value to the left of the decimal represents a date, and the value to the right of the decimal represents a time.

date expression

Any expression that can be interpreted as a date, including date literals, numbers that look like dates, strings that look like dates, and dates returned from functions. A date expression is limited to numbers or strings, in any combination, that can represent a date from January 1, 100 - December 31, 9999.

Dates are stored as part of a real number. Values to the left of the decimal represent the date; values to the right of the decimal represent the time. Negative numbers represent dates prior to December 30, 1899.

date literal

Any sequence of characters with a valid format that is surrounded by number signs (#). Valid formats include the date format specified by the locale settings for your code or the universal date format.

For example, #12/31/92# is the date literal that represents December 31, 1992, where English-U.S. is the locale setting for your application. Use date literals to maximize portability across national languages.

date separators

Characters used to separate the day, month, and year when date values are formatted. The characters are determined by system settings or by the **Format** function.

DBCS

A character set that uses 1 or 2 bytes to represent a character, allowing more than 256 characters to be represented.

dynamic data exchange (DDE)

An established protocol for exchanging data through active links between applications that run under Microsoft Windows.

Decimal data type

A data type that contains decimal numbers scaled by a power of 10. For zero-scaled numbers, that is, numbers with no decimal places, the range is +/-79,228,162,514,264,337,593,543,950,335. For numbers with 28 decimal places the range is +/-7.9228162514264337593543950335. The smallest non-zero number that can be represented as a **Decimal** is 0.000000000000000000000000000001.

Note that at this time the **Decimal** data type can only be used within a **Variant**. You cannot declare a variable to be of type **Decimal**. You can, however, create a **Variant** whose subtype is **Decimal** using the **CDec** function.

declaration

Nonexecutable code that names a constant, variable, or procedure, and specifies its characteristics, such as data type. For DLL procedures, declarations specify names, libraries, and arguments.

designer

Provides a visual design window in the Visual Basic development environment. You can use this window to design new classes visually. Visual Basic has built-in designers for forms. The Professional and Enterprise editions of Visual Basic include designers for ActiveX controls and ActiveX documents.

design time

The time during which you build an application in the development environment by adding controls, setting control or form properties, and so on. In contrast, during run time, you interact with the application like a user.

development environment

The part of the application where you write code, create controls, set control and form properties, and so on. This contrasts with running the application.

dynamic-link library (DLL)

A library of routines loaded and linked into applications at run time. DLLs are created with other programming languages such as C, MASM, or FORTRAN.

docked window

A window that is attached to the frame of the main window.

document

Any self-contained work created with an application and given a unique file name.

Double data type

A data type that holds double-precision floating-point numbers as 64-bit numbers in the range $-1.79769313486231E308$ to $-4.94065645841247E-324$ for negative values; $4.94065645841247E-324$ to $1.79769313486232E308$ for positive values. The number sign (#) type-declaration character represents the **Double** in Visual Basic.

E

[Back to top](#)

Empty

Indicates that no beginning value has been assigned to a **Variant** variable. An **Empty** variable is represented as 0 in a numeric context or a zero-length string ("") in a string context.

error number

A whole number in the range 0 - 65,535 that corresponds to the **Number** property setting of the **Err** object. When combined with the **Description** property setting of the **Err** object, this number represents a particular error message.

event source object

An object that is the source of events that occur in response to an action. An event source object is returned by a property. For example, the **CommandBarEvents** property returns the **CommandBarEvents** object.

executable file

A Windows-based application that can run outside the development environment. An executable file has an .exe file name extension.

expression

A combination of keywords, operators, variables, and constants that yields a string, number, or object. An expression can be used to perform a calculation, manipulate characters, or test data.

F

[Back to top](#)

file number

Number used in the **Open** statement to open a file. Use file numbers in the range 1 - 255, inclusive, for files not accessible to other applications. Use file numbers in the range 256 - 511 for files accessible from other applications.

focus

The ability to receive mouse clicks or keyboard input at any one time. In the Microsoft Windows environment, only one window, form, or control can have this ability at a time. The object that "has the focus" is normally indicated by a highlighted caption or title bar. The focus can be set by the user or by the application.

form

A window or dialog box. Forms are containers for controls. A multiple-document interface (MDI) form can also act as a container for child forms and some controls.

form module

A file in a Visual Basic project with an .frm file name extension that can contain graphical descriptions of a form; its controls and their property settings; form-level declarations of constants, variables, and external procedures; and event and general procedures.

Function procedure

A procedure that performs a specific task within a program and returns a value. A **Function** procedure begins with a **Function** statement and ends with an **End Function** statement.

G

[Back to top](#)

general procedure

A procedure that must be explicitly called by another procedure. In contrast, an event procedure is invoked automatically in response to a user or system action.

graphics method

A method that operates on an object such as a **Form**, **PictureBox**, or **Printer**, and performs run-time drawing operations such as animation or simulation. The graphics methods are **Circle**, **Cls**, **Line**, **PaintPicture**, **Point**, **Print**, and **PSet**.

H

[Back to top](#)

host application

Any application that supports the use of Visual Basic for Applications, for example, Microsoft Excel, Microsoft Project, and so on.

I

[Back to top](#)

icon

A graphical representation of an object or concept; commonly used to represent minimized applications in Microsoft Windows. An icon is a bitmap with a maximum size of 32 x 32 pixels. Icons have an .ico file name extension.

identifier

An element of an expression that refers to a constant or variable.

in process

Running in the same address space as an application.

insertable object

An application object that is a type of custom control, such as a Microsoft Excel worksheet.

Integer data type

A data type that holds integer variables stored as 2-byte whole numbers in the range -32,768 to 32,767. The **Integer** data type is also used to represent enumerated values. The percent sign (%) type-declaration character represents an **Integer** in Visual Basic.

intrinsic constants

A constant provided by an application. Visual Basic constants are listed in the object library and can be viewed with the **Object Browser**. Because you can't disable intrinsic constants, you can't create a user-defined constant with the same name.

K

[Back to top](#)

keyword

A word or symbol recognized as part of the Visual Basic programming language; for example, a statement, function name, or operator.

L

[Back to top](#)

line-continuation character

The combination of a space followed by an underscore (`_`) used in the development environment to extend a single logical line of code to two or more physical lines. However, you can't use a line-continuation character to continue a line of code within a string expression.

line label

Used to identify a single line of code. A line label can be any combination of characters that starts with a letter and ends with a colon (:). Line labels are not case sensitive and must begin in the first column.

line number

Used to identify a single line of code. A line number can be any combination of digits that is unique within the module where it is used. Line numbers must begin in the first column.

linked window

A window that is joined to another window other than the main window.

linked window frame

A window frame containing multiple windows that have been linked together.

locale

The set of information that corresponds to a given language and country.

The code locale setting affects the language of terms such as keywords and defines locale-specific settings such as the decimal and list separators, date formats, and character sorting order.

The system locale setting affects the way locale-aware functionality behaves, for example, when you display numbers or convert strings to dates. You set the system locale using the **Control Panel** utilities provided by the operating system. Although the code locale and system locale are generally set to the same setting, they may differ in some situations. For example, in Visual Basic, Standard Edition and Visual Basic, Professional Edition, the code is not translated from English-U.S. The system locale can be set to the user's language and country, but the code locale is always set to English-U.S. and can't be changed. In this case, the English-U.S. separators, format placeholders, and sorting order are used.

logic error

A programming error that can cause code to produce incorrect results or stop execution. For example, a logic error can be caused by incorrect variable names, incorrect variable types, endless loops, flaws in comparisons, or array problems.

Long data type

A 4-byte integer ranging in value from -2,147,483,648 to 2,147,483,647.

The ampersand (&) type-declaration character represents a **Long** in Visual Basic.

M

[Back to top](#)

margin indicator

An icon displayed in the **Margin Indicator** bar in the **Code** window.

Margin indicators provide visual cues during code editing.

MDI child

A form contained within an MDI form in a multiple-document interface (MDI) application. To create a child form, set the **MDIChild** property of the MDI form to **True**.

MDI form

A window that makes up the background of a multiple-document interface (MDI) application. The MDI form is the container for any MDI child forms in the application.

member

An element of a collection, object, or user-defined type.

metafile

A file that stores an image as graphical objects such as lines, circles, and polygons rather than as pixels. There are two types of metafiles, standard and enhanced. Standard metafiles usually have a .wmf file name extension. Enhanced metafiles usually have a .emf file name extension. Metafiles preserve an image more accurately than pixels when the image is resized.

method

A procedure that acts on an object.

module

A set of declarations followed by procedures.

module level

Describes code in the Declarations section of a module. Any code outside a procedure is referred to as module-level code. Declarations must be listed first, followed by procedures.

module variable

A variable declared outside **Function**, **Sub**, or **Property** procedure code. Module variables must be declared outside any procedures in the module. They exist while the module is loaded and are visible in all procedures in the module.

N

[Back to top](#)

named argument

An argument that has a name that is predefined in the object library. Instead of providing a value for each argument in a specified order expected by the syntax, you can use named arguments to assign values in any order. For

example, suppose a method accepts three arguments:

DoSomething *namedarg1, namedarg2, namedarg3* By assigning values to named arguments, you can use the following statement: **DoSomething** *namedarg3 := 4, namedarg2 := 5, namedarg1 := 20* Note that the named arguments don't have to appear in the normal positional order in the syntax.

Null

A value indicating that a variable contains no valid data. **Null** is the result of an explicit assignment of **Null** to a variable or any operation between expressions that contain **Null**.

numeric data type

Any intrinsic numeric data type (**Byte, Boolean, Integer, Long, Currency, Single, Double, or Date**).

numeric expression

Any expression that can be evaluated as a number. Elements of an expression can include any combination of keywords, variables, constants, and operators that result in a number.

numeric type

Any intrinsic numeric data type (**Byte, Boolean, Integer, Long, Currency, Single, Double, or Date**) or any **Variant** numeric subtype (**Empty, Integer, Long, Single, Double, Currency, Decimal, Date, Error, Boolean, or Byte**).

O

[Back to top](#)

object

A combination of code and data that can be treated as a unit, for example, a control, form, or application component. Each object is defined by a class.

Object box

A list box at the upper-left corner of the **Code** window that lists the form and controls in the form to which the code is attached, or a list box located at the top of the **Properties** window that lists the form and its controls.

Object Browser

A dialog box in which you can examine the contents of an object library to get information about the objects provided.

Object data type

A data type that represents any **Object** reference. **Object** variables are stored as 32-bit (4-byte) addresses that refer to objects.

object expression

An expression that specifies a particular object and can include any of the object's containers. For example, an application can have an **Application** object that contains a **Document** object that contains a **Text** object.

object library

A file with the .olb extension that provides information to Automation controllers (like Visual Basic) about available objects. You can use the **Object Browser** to examine the contents of an object library to get information about the objects provided.

object module

A module that contains code specific to an object, for example, class module, form module, and document module. Object modules contain the code behind their associated objects. The rules for object modules differ from those for standard modules.

object type

A type of object exposed by an application through Automation, for example, **Application**, **File**, **Range**, and **Sheet**. Use the **Object Browser** or refer to the application's documentation for a complete listing of available objects.

object variable

A variable that contains a reference to an object.

P

[Back to top](#)

parameter

Variable name by which an argument passed to a procedure is known within the procedure. This variable receives the argument passed into the procedure. Its scope ends when the procedure ends.

path

A string expression specifying a directory or folder location. The location can include a drive specification.

pi

A mathematical constant equal to approximately 3.1415926535897932.

point

A point is 1/72 inch. Font sizes are usually measured in points.

print zone

Print zones begin every 14 columns. The width of each column is an

average of the width of all characters in the point size for the selected font.

Private

Variables that are visible only to the module in which they are declared.

procedure

A named sequence of statements executed as a unit. For example, **Function**, **Property**, and **Sub** are types of procedures. A procedure name is always defined at module level. All executable code must be contained in a procedure. Procedures can't be nested within other procedures.

Procedure box

A list box at the upper-right corner of the **Code** window and the **Debug** window that displays the procedures recognized for the object displayed in the **Object** box.

procedure call

A statement in code that tells Visual Basic to execute a procedure.

procedure level

Describes statements located within a **Function**, **Property**, or **Sub** procedure. Declarations are usually listed first, followed by assignments and other executable code. Note that module-level code resides outside a procedure block.

project

A set of modules.

Project window

A window that displays a list of the form, class, and standard modules; the resource file; and references in your project. Files with .ocx and .vbx file name extensions aren't displayed in the **Project** window.

Properties window

A window used to display or change properties of a selected form or control at design time. Some custom controls have customized **Properties** windows.

property

A named attribute of an object. Properties define object characteristics such as size, color, and screen location, or the state of an object, such as enabled or disabled.

Property procedure

A procedure that creates and manipulates properties for a class module. A **Property** procedure begins with a **Property Let**, **Property Get**, or **Property Set** statement and ends with an **End Property** statement.

Public

Variables declared using the **Public** statement are visible to all procedures

in all modules in all applications unless **Option Private Module** is in effect. In that case, the variables are public only within the project in which they reside.

R

[Back to top](#)

referenced project

The project you directly create a link to from the current project you are working on. A project referenced by one of the current project's directly referenced projects is called an indirectly referenced project. Its **Public** variables are not accessible to the current project except through qualification with its project name. Any combination of direct and indirect references between projects is valid as long as they do not result in a complete cycle.

referencing project

The current project. How you create a link to a project depends on the host application. For example, to directly reference a project in Microsoft Excel, select the project from the **References** dialog box of the **Tools** menu. **Public** variables in a directly referenced project are visible to the directly referencing project, but **Public** variables in a directly referencing project are not visible to a directly referenced project.

registry

In Microsoft Windows version 3.1, OLE registration information and file associations are stored in the registration database, and program settings are stored in Windows system initialization (.ini) files. In Microsoft Windows 95, the Windows registry serves as a central configuration database for user, application, and computer-specific information, including the information previously contained in both the Windows version 3.1 registration database and .ini files.

resource file

A file in a Visual Basic project with an .res file name extension that can contain bitmaps, text strings, or other data. By storing this data in a separate file, you can change the information without editing your code. Only one resource file can be associated with a project.

run-time error

An error that occurs when code is running. A run-time error results when a statement attempts an invalid operation.

run time

The time during which code is running. During run time, you can't edit the code.

S

[Back to top](#)

scope

Defines the visibility of a variable, procedure, or object. For example, a variable declared as **Public** is visible to all procedures in all modules in a directly referencing project unless **Option Private Module** is in effect. When **Option Private Module** is in effect, the module itself is private and therefore not visible to referencing projects. Variables declared in a procedure are visible only within the procedure and lose their value between calls unless they are declared **Static**.

seed

An initial value used to generate pseudorandom numbers. For example, the **Randomize** statement creates a seed number used by the **Rnd** function to create unique pseudorandom number sequences.

Single data type

A data type that stores single-precision floating-point variables as 32-bit (4-byte) floating-point numbers, ranging in value from -3.402823E38 to -1.401298E-45 for negative values, and 1.401298E-45 to 3.402823E38 for positive values. The exclamation point (!) type-declaration character represents a **Single** in Visual Basic.

sort order

A sequencing principle used to order data, for example, alphabetic, numeric, ascending, descending, and so on.

stack

A fixed amount of memory used by Visual Basic to preserve local variables and arguments during procedure calls.

standard module

A module containing only procedure, type, and data declarations and definitions. Module-level declarations and definitions in a standard module are **Public** by default. A standard module is referred to as a code module in earlier versions of Visual Basic.

statement

A syntactically complete unit that expresses one kind of action, declaration,

or definition. A statement generally occupies a single line, although you can use a colon (:) to include more than one statement on a line. You can also use a line-continuation character (_) to continue a single logical line onto a second physical line.

string comparison

A comparison of two sequences of characters. Use **Option Compare** to specify binary or text comparison. In English-U.S., binary comparisons are case sensitive; text comparisons are not.

string constant

Any constant (defined using the **Const** keyword) consisting of a sequence of contiguous characters interpreted as the characters themselves rather than as a numeric value.

String data type

A data type consisting of a sequence of contiguous characters that represent the characters themselves rather than their numeric values. A **String** can include letters, numbers, spaces, and punctuation. The **String** data type can store fixed-length strings ranging in length from 0 to approximately 63K characters and dynamic strings ranging in length from 0 to approximately 2 billion characters. The dollar sign (\$) type-declaration character represents a **String** in Visual Basic.

string expression

Any expression that evaluates to a sequence of contiguous characters. Elements of a string expression can include a function that returns a string, a string literal, a string constant, a string variable, a string **Variant**, or a function that returns a string **Variant (VarType 8)**.

string literal

Any expression consisting of a sequence of contiguous characters surrounded by quotation marks that is literally interpreted as the characters within the quotation marks.

Sub procedure

A procedure that performs a specific task within a program, but returns no explicit value. A **Sub** procedure begins with a **Sub** statement and ends with an **End Sub** statement.

syntax checking

A feature that checks your code for correct syntax. If the syntax checking feature is enabled, a message is displayed when you enter code that contains a syntax error and the suspect code is highlighted.

syntax error

An error that occurs when you enter a line of code that Visual Basic doesn't

recognize.

Note that syntax rules for individual keywords are defined in the Syntax section of the associated Help topic. To get Help on a keyword from within the development environment, select the keyword and press F1.

T

[Back to top](#)

tab order

The order in which the focus moves from one field to the next as you press TAB or SHIFT+TAB.

time expression

Any expression that can be interpreted as a time. This includes any combination of time literals, numbers that look like times, strings that look like times, and times returned from functions.

Times are stored as part of a real number. Values to the right of the decimal represent the time. For example, midday (12:00 P.M.) is represented by 0.5.

twip

A unit of screen measurement equal to 1/20 point. A twip is a screen-independent unit used to ensure that placement and proportion of screen elements in your screen application are the same on all display systems. There are approximately 1440 twips to a logical inch or 567 twips to a logical centimeter (the length of a screen item measuring one inch or one centimeter when printed).

type-declaration character

A character appended to a variable name indicating the variable's data type. By default, variables are of type **Variant** unless a corresponding **Def type** statement is present in the module.

type library

A file or component within another file that contains standard descriptions of exposed objects, properties, and methods that are available for Automation. Object library files (.olb) contain type libraries.

U

[Back to top](#)

Unicode

International Standards Organization (ISO) character standard. Unicode

uses a 16-bit (2-byte) coding scheme that allows for 65,536 distinct character spaces. Unicode includes representations for punctuation marks, mathematical symbols, and dingbats, with substantial room for future expansion.

universal date format

The universal date format is #yyyy-mm-dd hh:mm:ss#. However, both the date component (#yyyy-mm-dd#) and the time component (#hh:mm:ss#) can be represented separately.

user-defined type

Any data type defined using the **Type** statement. User-defined data types can contain one or more elements of any data type. Arrays of user-defined and other data types are created using the **Dim** statement. Arrays of any type can be included within user-defined types.

V

[Back to top](#)

variable

A named storage location that can contain data that can be modified during program execution. Each variable has a name that uniquely identifies it within its scope. A data type can be specified or not.

Variable names must begin with an alphabetic character, must be unique within the same scope, can't be longer than 255 characters, and can't contain an embedded period or type-declaration character.

Variant data type

A special data type that can contain numeric, string, or date data as well as user-defined types and the special values **Empty** and **Null**. The **Variant** data type has a numeric storage size of 16 bytes and can contain data up to the range of a **Decimal**, or a character storage size of 22 bytes (plus string length), and can store any character text. The **VarType** function defines how the data in a **Variant** is treated. All variables become **Variant** data types if not explicitly declared as some other data type.

variant expression

Any expression that can evaluate to numeric, string, or date data, as well as the special values **Empty** and **Null**.

W

[Back to top](#)

watch expression

A user-defined expression that enables you to observe the behavior of a variable or expression. Watch expressions appear in the Watch window of the Visual Basic Editor and are automatically updated when you enter break mode. The **Watch** window displays the value of an expression within a given context. Watch expressions are not saved with your code.

Z

[Back to top](#)

z-order

The visual layering of controls on a form along the form's z-axis (depth). The z-order determines which controls are in front of other controls.

[Show All](#)

Create an expression

You create an expression by combining [identifiers](#), [operators](#), and values to produce a result. You can either create the expression on your own or use the Expression Builder to help you create the expression.



[Create an expression on your own](#)

In the control where you want the result, enter a combination of identifiers, operators, and values. For example, the following expression increases the value displayed in the Freight control on the Orders form by 10 percent:

```
= [Forms]![Orders]![Freight] * 1.1
```

Notes

- Use the ! operator in an expression if the item that follows is user-defined (an element of a [collection](#)).
- Use the . (dot) operator if the item that follows is defined by Microsoft Access.
- Always precede [calculated control](#) expressions by an equal sign (=).
- Some expressions produce a true or false result. For example, if you enter this type of expression in the **Condition** column of a macro, Microsoft Access carries out the specified [action](#) only when the expression evaluates to true.



[Use the Expression Builder](#)

1. Start the Expression Builder.



[How?](#)

You can start the Expression Builder from most places where you write [expressions](#), such as in a [property sheet](#), a **Criteria** cell in the query [design](#)

[grid](#), or the [Macro window](#).

[In a property sheet or the lower pane of the Macro window](#)

1. Do one of the following:

[Start the Expression Builder by opening a property sheet](#)

- To set properties for a form or report, double-click the [form selector](#) or the [report selector](#).
- To set properties for the data access page, click **Select Page** on the **Edit** menu, and then click **Properties** on the toolbar.
- To set properties for a table or query, click **Properties** on the toolbar.
- To set properties for a section on a form or report, double-click the [section selector](#).
- To set properties for a section on a data access page, double-click the [section bar](#).
- To set properties of a query field, click the cell in the **Field** row, and then click **Properties** on the toolbar.
- To set properties of a query field list, click anywhere in the list, and then click **Properties** on the toolbar.
- To set properties for a [control](#), click the control, and then click **Properties** on the toolbar.

[Open a macro in macro Design view](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
 2. Click the name of the macro you want to open.
 3. Click the **Design** button on the Database window toolbar.
2. Click the property or [action argument](#) box that will contain the expression.
 3. Click the **Build** button next to the property or argument box.

If a property has more than one builder associated with it, Microsoft Access displays the **Choose Builder** dialog box. Click **Expression**

Builder and click **OK**.

Note If the property box or argument box where you start the Expression Builder already contains a value, that value is automatically copied into the expression box.

 [In the query design grid or a macro condition](#)

You can start the Expression Builder in a **Field** or **Criteria** cell in query [Design view](#) or in the **Condition** column of the Macro window.

1. Right-click where you want to insert the expression.
2. On the [shortcut menu](#), click **Build**.

Note If the cell or **Condition** column where you start the Expression Builder already contains a value, that value is automatically copied into the expression box.

2. In the lower-left box of the Expression Builder, double-click or click the folder containing the element you want.
3. In the lower-middle folder, double-click an element to paste it into the expression box, or click a category of elements.
4. If you select a category in the lower-middle box, values display in the lower-right box. Double-click a value to paste it in the expression box.

 [Tip](#)

You can also type any part of the expression directly in the expression box.

5. Paste any operators you want in the expression by placing the insertion point in the expression box where you want the operator, and clicking one of the operator buttons that are in the middle of the builder.
6. When your expression is complete, click **OK**.

Microsoft Access copies your expression to the location where you started the Expression Builder. If that location already contains a value, your new expression replaces the value or text.

[Show All](#)

Refer to objects in expressions

An [expression](#) is a formula or calculation that you can use to obtain a result. Frequently, you must refer to objects in your database from an expression to return a value.

For example, you might need to refer to:

[The value of a control or property](#)

You refer to the value of a [control](#) or property in an [expression](#) by typing its [identifier](#).

To refer to the value of a control

- Type the identifier for the form or report that contains the control, followed by the ! operator and the name of the control.

For example, the following identifier refers to the value of the OrderID control on the Orders form:

Forms![Orders]![OrderID]

[Tip](#)

If you're referring to a control on a subform or subreport, you don't have to specify the full identifier for the form or report using the Form or Report property. For example, you can use the following identifier to refer to the Quantity control on the Orders Subform subform:

Forms![Orders]![Orders Subform]![Quantity]

To refer to the value of a property

- Type the identifier for the form, report, or control to which the property applies, followed by the . (dot) operator, and the name of the property.

For example, the following identifier refers to the **DefaultValue** property of the OrderDate control on the Orders form:

Forms![Orders]![OrderDate].DefaultValue

Notes

- You can use the [Expression Builder](#) to create an identifier for a control or property.
- When you run a macro or [Microsoft Visual Basic for Applications](#) code containing an expression that refers to a form or report, the form or report must be open.

 [A form, report, subform, or subreport](#)

You refer to an open form or report, or the subform or subreport of an open form or report, in an [expression](#) by typing its [identifier](#).

To refer to an open form or report

- Type the name of the **Forms** or **Reports** [collection](#), followed by the ! operator and the name of the form or report.

For example, the following identifier refers to the Orders form:

Forms![Orders]

To refer to a subform or subreport

Refer to the subform or subreport [control](#) on the form or report that contains the subform or subreport, then use the **Form** or **Report** property of the control to refer to the actual subform or subreport.

- Type the identifier for the form or report that contains the subform or subreport, followed by the name of its subform or subreport control, the . (dot) operator, and the **Form** or **Report** property.

For example, the following identifier refers to the Orders Subform subform on the Orders form:

Forms![Orders]![Orders Subform].Form

Notes

- You can use the Expression Builder to create an identifier for a form, report, subform, or subreport.
- When you run a macro or [Microsoft Visual Basic for Applications](#) code containing an expression that refers to a form or report, the form or report must be open.

[A form or report section or report group level](#)

- Type the [identifier](#) for the form or report, followed by the . (dot) operator and the name of the section or group level.

For example, the following identifier refers to the [detail section](#) on the Orders form:

[Forms]![Orders].Detail

Notes

- When you run a macro or [Microsoft Visual Basic for Applications](#) code containing an [expression](#) that refers to a form or report, the form or report must be open.
- You can also refer to form or report sections or report group levels by index number using the **Section** property.

[A column in a list](#)

You can refer to a column in a multiple-column list box or combo box by using the **Column** property of the list box or combo box [control](#).

- Type the [identifier](#) for the **Column** property, followed by the index number of the column in parentheses. Use 0 (zero) to refer to the first column, 1 to refer to the second column, and so on.

For example, the following identifier refers to the second column of the Country list box control on the Customers form:

[Forms]![Customers]![Country].Column(1)

Note When you run a macro or [Microsoft Visual Basic for Applications](#) code containing an [expression](#) that refers to a form or report, the form or report must be open.

[Show All](#)

About referring to an object or its properties in expressions

You use an [identifier](#) in an [expression](#) to refer to an object or its properties. For example, you can refer to an open form, an open report, a control on an open form or report, or any properties of the form or report.

 [Refer to an object by using a full identifier](#)

The following identifier refers to the **Visible** property of a control:

Reports![Invoice]![ShipName].Visible

The full identifier for an object or property shows the relationship between items in the identifier. In this identifier:

- Reports refers to the [collection](#) of open reports in the database. Microsoft Access automatically creates **Forms** and **Reports** collections for each database. The **Forms** collection is made up of all open forms, while the **Reports** collection is made up of all open reports.
- [Invoice] refers to the Invoice report in the **Reports** collection.
- [ShipName] refers to the ShipName control on the Invoice report.
- Visible refers to the **Visible** property of the ShipName control.

It's a good idea to refer to an object or property using its full identifier. In some cases, a full identifier is required. For example, to refer to a control on a form or report that isn't the form or report with the focus, you must type its full identifier. The following expression displays the sum of the values in the Subtotal and Freight controls on the Orders form in a control on a different form:

= Forms![Orders]![Subtotal] + Forms![Orders]![Freight]

 [Refer to an object without a full identifier](#)

In some circumstances, you can refer to a control or its properties without specifying a full identifier:

- If you're referring to a control on the current form or report, you don't have to specify the form or report identifier. For example, to display the sum of the values in the Subtotal and Freight controls in a different control on the same form, set the **ControlSource** property of the control to:

= [Subtotal] + [Freight]

- If you're referring to a control on a subform or subreport, you don't have to specify the full identifier for the form or report using the **Form** or **Report** property. For example, you can use the following identifier to refer to the Quantity control on the Orders Subform subform:

Forms![Orders]![Orders Subform]![Quantity]

The full identifier for the Quantity control would be:

Forms![Orders]![Orders Subform].Form![Quantity]

- In a macro or [action argument](#), you don't have to specify the identifier for the form or report from which the macro is run. For example, if you set an [event property](#) on a form to the name of a macro, you can refer to controls on the form in the macro's **Condition** column or action arguments without specifying the form's identifier.
- In a [Microsoft Visual Basic for Applications](#) procedure, you can use the **Me** keyword rather than the full identifier to refer to a control on the current form or report. For example, to assign the sum of the values in the Subtotal and Freight controls on a form to the variable OrderTotal in one of the form's event procedures, add the following [statement](#) to the event procedure:

```
OrderTotal = Me![Subtotal] + Me![Freight]
```

Notes

- When you run a macro or [Microsoft Visual Basic for Applications](#) code containing an expression that refers to a form or report, the form or report must be open.
- In a [Microsoft Visual Basic](#) procedure, you can refer to an object by enclosing its name in parentheses and double (") quotation marks instead of

using the ! operator. The parentheses syntax is required if you want to use a variable in an identifier. For example, the following identifiers are equivalent:

```
Forms![Orders]![OrderDate]  
Forms("Orders")("OrderDate")
```

[Show All](#)

Enable the use of an ActiveX control in expressions

By default, Microsoft Access does not allow you to use an [ActiveX control](#) in expressions. It is possible, however, to enable the use of your ActiveX control in expressions when the ActiveX control has been added to a form. Follow these steps to enable the use of an ActiveX control in an expression:

1. In the Database window, click **Modules**  under **Objects**, and then click **New** on the Database window toolbar.
2. Type the following code into the module:

```
Sub EnableActiveXControl()  
    SysCmd 14, "<ActiveX Control GUID>"  
End Sub
```

Note Replace <ActiveX Control GUID> with the globally unique identifier (GUID) that identifies the ActiveX control that you want to enable in expressions.

3. Press F5 to run the procedure.

The ActiveX control is added to the list of that Access allows to be used in expressions.

Notes

- You cannot remove an ActiveX control once it has been added to the list of allowed controls.
- Access allows the use of an ActiveX control in expressions only when the control has been added to a form and the ActiveX control has been loaded by the form. The ActiveX control cannot refer to a property or method of an ActiveX control that has not been added to the form.

[Show All](#)

Examples of referring to controls in expressions

[Referring to controls on a form or a subform](#)

The following table lists examples of expressions you can use in calculated controls on forms.

If you use this expression	Microsoft Access displays
=Forms![Orders]! [OrderID]	The value of the OrderID control on the Orders form.
=Forms![Orders]! [Orders Subform]! [OrderSubtotal]	The value of the OrderSubtotal control on the Orders Subform on the Orders form.
=Forms![Orders]! [Orders Subform]! [ProductID].Column(2)	The value of the third column in ProductID, a multiple-column list box on the Orders Subform on the Orders form. (0 refers to the first column, 1 refers to the second, and so on.)
=Forms![Orders]! [Orders Subform]! [Price] * 1.06	The product of the value of the Price control on the Orders Subform on the Orders form and 1.06 (adds 6 percent to the value of the Price control).
=Parent![OrderID]	The value of the OrderID control on the main or parent form of the current subform.

[Referring to controls on a report or a subreport](#)

The following table lists examples of expressions you can use in calculated controls on reports.

If you use this expression	Microsoft Access displays
=Reports![Invoice]![OrderID]	The value of the OrderID control on the Invoice report.

=Reports![Summary]!
[Summary Subreport]!
[SalesTotal]

The value of the SalesTotal control on the Summary Subreport on the Summary report.

=Parent![OrderID]

The value of the OrderID control on the main or parent report of the current subreport.

 [Referring to controls on a data access page](#)

The following table lists examples of expressions that you can use in calculated controls on data access pages.

If you use this expression	The page displays
<code>=Document.All("MSODSC").CurrentSection .ChildSection.HTMLContainer.Children("ProductID")</code>	The value in the ProductID control in the section that is one group level below the current section.
<code>=Document.All("MSODSC").CurrentSection .ParentSection.HTMLContainer.Children("CustomerID")</code>	The value in the CustomerID control in the section that is one group level above the current section.
<code>=Document.All("MSODSC").CurrentSection .Parent.Parent.HTMLContainer.Children("Country")</code>	The value in the Country control in the section that is two group levels above the current section.
<code>=Document.All("MySpreadsheet").Range("A1")</code>	The values in the range A1 in the spreadsheet control.

The following table lists examples of referring to controls on a page from a spreadsheet control.

If you use this expression	The page displays
<code>=document.Quantity.value*document.UnitPrice.value</code>	The result of multiplying the value in the Quantity

=C\$1*document.UnitPrice.value

control by the value in the UnitPrice control.

The result of multiplying the value in the UnitPrice control by a percentage rate that is stored in spreadsheet cell C1.

Notes

- In a calculated control, precede the expression with the equal sign (=) operator.
- When you set the **Name** property of a calculated control, make sure you use a unique name. Don't use the name of one of the controls you used in the expression.

LBound Function

Returns a Long containing the smallest available subscript for the indicated dimension of an array.

Syntax

LBound(*arrayname* [, *dimension*])

The **LBound** function syntax has these parts:

Part	Description
<i>arrayname</i>	Required. Name of the array variable; follows standard variable naming conventions.
<i>dimension</i>	Optional; Variant (Long) . Whole number indicating which dimension's lower bound is returned. Use 1 for the first dimension, 2 for the second, and so on. If <i>dimension</i> is omitted, 1 is assumed.

Remarks

The **LBound** function is used with the **UBound** function to determine the size of an array. Use the **UBound** function to find the upper limit of an array dimension.

LBound returns the values in the following table for an array with the following dimensions:

```
Dim A(1 To 100, 0 To 3, -3 To 4)
```

Statement	Return Value
LBound(A, 1)	1
LBound(A, 2)	0
LBound(A, 3)	-3

The default lower bound for any dimension is either 0 or 1, depending on the

setting of the **Option Base** statement. The base of an array created with the **Array** function is zero; it is unaffected by **Option Base**.

Arrays for which dimensions are set using the **To** clause in a **Dim**, **Private**, **Public**, **ReDim**, or **Static** statement can have any integer value as a lower bound.

UBound Function

Returns a Long containing the largest available subscript for the indicated dimension of an array.

Syntax

UBound(*arrayname*[, *dimension*])

The **UBound** function syntax has these parts:

Part	Description
<i>arrayname</i>	Required. Name of the array variable; follows standard variable naming conventions.
<i>dimension</i>	Optional; Variant (Long) . Whole number indicating which dimension's upper bound is returned. Use 1 for the first dimension, 2 for the second, and so on. If <i>dimension</i> is omitted, 1 is assumed.

Remarks

The **UBound** function is used with the **LBound** function to determine the size of an array. Use the **LBound** function to find the lower limit of an array dimension.

UBound returns the following values for an array with these dimensions:

```
Dim A(1 To 100, 0 To 3, -3 To 4)
```

Statement	Return Value
UBound(A, 1)	100
UBound(A, 2)	3
UBound(A, 3)	4

Asc Function

Returns an Integer representing the character code corresponding to the first letter in a string.

Syntax

Asc(*string*)

The required *string* argument is any valid string expression . If the *string* contains no characters, a run-time error occurs.

Remarks

The range for returns is 0 – 255 on non-DBCS systems, but –32768 – 32767 on DBCS systems.

Note The **AscB** function is used with byte data contained in a string. Instead of returning the character code for the first character, **AscB** returns the first byte. The **AscW** function returns the Unicode character code except on platforms where Unicode is not supported, in which case, the behavior is identical to the **Asc** function.

Note Visual Basic for the Macintosh does not support Unicode strings. Therefore, **AscW**(*n*) cannot return all Unicode characters for *n* values in the range of 128 – 65,535, as it does in the Windows environment. Instead, **AscW**(*n*) attempts a "best guess" for Unicode values *n* greater than 127. Therefore, you should not use **AscW** in the Macintosh environment.

Chr Function

Returns a String containing the character associated with the specified character code .

Syntax

Chr(*charcode*)

The required *charcode* argument is a Long that identifies a character.

Remarks

Numbers from 0 – 31 are the same as standard, nonprintable ASCII codes. For example, **Chr**(10) returns a linefeed character. The normal range for *charcode* is 0 – 255. However, on DBCS systems, the actual range for *charcode* is -32768 to 65535.

Note The **ChrB** function is used with byte data contained in a **String**. Instead of returning a character, which may be one or two bytes, **ChrB** always returns a single byte. The **ChrW** function returns a **String** containing the Unicode character except on platforms where Unicode is not supported, in which case, the behavior is identical to the **Chr** function.

Note Visual Basic for the Macintosh does not support Unicode strings. Therefore, **ChrW**(*n*) cannot return all Unicode characters for *n* values in the range of 128 – 65,535, as it does in the Windows environment. Instead, **ChrW**(*n*) attempts a "best guess" for Unicode values *n* greater than 127. Therefore, you should not use **ChrW** in the Macintosh environment.

DateSerial Function

Returns a **Variant (Date)** for a specified year, month, and day.

Syntax

DateSerial(*year, month, day*)

The **DateSerial** function syntax has these named arguments:

Part	Description
<i>year</i>	Required; Integer . Number between 100 and 9999, inclusive, or a numeric expression.
<i>month</i>	Required; Integer . Any numeric expression.
<i>day</i>	Required; Integer . Any numeric expression.

Remarks

To specify a date, such as December 31, 1991, the range of numbers for each **DateSerial** argument should be in the accepted range for the unit; that is, 1–31 for days and 1–12 for months. However, you can also specify relative dates for each argument using any numeric expression that represents some number of days, months, or years before or after a certain date.

The following example uses numeric expressions instead of absolute date numbers. Here the **DateSerial** function returns a date that is the day before the first day (1 - 1), two months before August (8 - 2), 10 years before 1990 (1990 - 10); in other words, May 31, 1980.

```
DateSerial(1990 - 10, 8 - 2, 1 - 1)
```

Under Windows 2000, two digit years for the *year* argument are interpreted based on user-defined machine settings. The default settings are that values between 0 and 29, inclusive, are interpreted as the years 2000–2029. The default values between 30 and 99 are interpreted as the years 1930–1999. For all other

year arguments, use a four-digit year (for example, 1800).

Earlier versions of Windows interpret two-digit years based on the defaults described above. To be sure the function returns the proper value, use a four-digit year.

When any argument exceeds the accepted range for that argument, it increments to the next larger unit as appropriate. For example, if you specify 35 days, it is evaluated as one month and some number of days, depending on where in the year it is applied. If any single argument is outside the range -32,768 to 32,767, an error occurs. If the date specified by the three arguments falls outside the acceptable range of dates, an error occurs.

Note For *year*, *month*, and *day*, if the **Calendar** property setting is Gregorian, the supplied value is assumed to be Gregorian. If the **Calendar** property setting is Hijri, the supplied value is assumed to be Hijri.

The returned date part is in the time period units of the current Visual Basic calendar. For example, if the current calendar is Hijri and the date part to be returned is the year, the year value is a Hijri year. For the argument *year*, values between 0 and 99, inclusive, are interpreted as the years 1400-1499. For all other *year* values, use the complete four-digit year (for example, 1520).

DateValue Function

Returns a **Variant (Date)**.

Syntax

DateValue(*date*)

The required *date* argument is normally a string expression representing a date from January 1, 100 through December 31, 9999. However, *date* can also be any expression that can represent a date, a time, or both a date and time, in that range.

Remarks

If *date* is a string that includes only numbers separated by valid date separators, **DateValue** recognizes the order for month, day, and year according to the Short Date format you specified for your system. **DateValue** also recognizes unambiguous dates that contain month names, either in long or abbreviated form. For example, in addition to recognizing 12/30/1991 and 12/30/91, **DateValue** also recognizes December 30, 1991 and Dec 30, 1991.

If the year part of *date* is omitted, **DateValue** uses the current year from your computer's system date.

If the *date* argument includes time information, **DateValue** doesn't return it. However, if *date* includes invalid time information (such as "89:98"), an error occurs.

Note For *date*, if the **Calendar** property setting is Gregorian, the supplied date must be Gregorian. If the calendar is Hijri, the supplied date must be Hijri. If the supplied date is Hijri, the argument *date* is a **String** representing a date from 1/1/100 (Gregorian Aug 2, 718) through 4/3/9666 (Gregorian Dec 31, 9999).

Day Function

Returns a **Variant (Integer)** specifying a whole number between 1 and 31, inclusive, representing the day of the month.

Syntax

Day(*date*)

The required *date* argument is any Variant, numeric expression, string expression, or any combination, that can represent a date. If *date* contains Null, **Null** is returned.

Note If the **Calendar** property setting is Gregorian, the returned integer represents the Gregorian day of the month for the date argument. If the calendar is Hijri, the returned integer represents the Hijri day of the month for the date argument.

FormatCurrency Function

Description

Returns an expression formatted as a currency value using the currency symbol defined in the system control panel.

Syntax

FormatCurrency(*Expression* [, *NumDigitsAfterDecimal* [, *IncludeLeadingDigit* [, *UseParensForNegativeNumbers* [, *GroupDigits*]]]])

The **FormatCurrency** function syntax has these parts:

Part	Description
<i>Expression</i>	Required. Expression to be formatted.
<i>NumDigitsAfterDecimal</i>	Optional. Numeric value indicating how many places to the right of the decimal are displayed. Default value is -1, which indicates that the computer's regional settings are used.
<i>IncludeLeadingDigit</i>	Optional. Tristate constant that indicates whether or not a leading zero is displayed for fractional values. See Settings section for values.
<i>UseParensForNegativeNumbers</i>	Optional. Tristate constant that indicates whether or not to place negative values within parentheses. See Settings section for values.
<i>GroupDigits</i>	Optional. Tristate constant that indicates whether or not numbers are grouped using the group delimiter specified in the computer's regional settings. See Settings section for values.

Settings

The *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*

arguments have the following settings:

Constant	Value	Description
vbTrue	-1	True
vbFalse	0	False
vbUseDefault	-2	Use the setting from the computer's regional settings.

Remarks

When one or more optional arguments are omitted, the values for omitted arguments are provided by the computer's regional settings.

The position of the currency symbol relative to the currency value is determined by the system's regional settings.

Note All settings information comes from the **Regional Settings Currency** tab, except leading zero which comes from the **Number** tab.

FormatDateTime Function

Description

Returns an expression formatted as a date or time.

Syntax

FormatDateTime(*Date*[,*NamedFormat*])

The **FormatDateTime** function syntax has these parts:

Part	Description
<i>Date</i>	Required. Date expression to be formatted.
<i>NamedFormat</i>	Optional. Numeric value that indicates the date/time format used. If omitted, vbGeneralDate is used.

Settings

The *NamedFormat* argument has the following settings:

Constant	Value	Description
vbGeneralDate	0	Display a date and/or time. If there is a date part, display it as a short date. If there is a time part, display it as a long time. If present, both parts are displayed.
vbLongDate	1	Display a date using the long date format specified in your computer's regional settings.
vbShortDate	2	Display a date using the short date format specified in your computer's regional settings.
vbLongTime	3	Display a time using the time format specified in your computer's regional settings.
vbShortTime	4	Display a time using the 24-hour format (hh:mm).

FormatNumber Function

Description

Returns an expression formatted as a number.

Syntax

FormatNumber(*Expression* [, *NumDigitsAfterDecimal* [, *IncludeLeadingDigit* [, *UseParensForNegativeNumbers* [, *GroupDigits*]]]])

The **FormatNumber** function syntax has these parts:

Part	Description
<i>Expression</i>	Required. Expression to be formatted.
<i>NumDigitsAfterDecimal</i>	Optional. Numeric value indicating how many places to the right of the decimal are displayed. Default value is -1, which indicates that the computer's regional settings are used.
<i>IncludeLeadingDigit</i>	Optional. Tristate constant that indicates whether or not a leading zero is displayed for fractional values. See Settings section for values.
<i>UseParensForNegativeNumbers</i>	Optional. Tristate constant that indicates whether or not to place negative values within parentheses. See Settings section for values.
<i>GroupDigits</i>	Optional. Tristate constant that indicates whether or not numbers are grouped using the group delimiter specified in the computer's regional settings. See Settings section for values.

Settings

The *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits* arguments have the following settings:

Constant	Value	Description
vbTrue	-1	True
vbFalse	0	False
vbUseDefault	-2	Use the setting from the computer's regional settings.

Remarks

When one or more optional arguments are omitted, the values for omitted arguments are provided by the computer's regional settings.

Note All settings information comes from the **Regional Settings Number** tab.

FormatPercent Function

Description

Returns an expression formatted as a percentage (multiplied by 100) with a trailing % character.

Syntax

FormatPercent(*Expression* [, *NumDigitsAfterDecimal* [, *IncludeLeadingDigit* [, *UseParensForNegativeNumbers* [, *GroupDigits*]]]])

The **FormatPercent** function syntax has these parts:

Part	Description
<i>Expression</i>	Required. Expression to be formatted.
<i>NumDigitsAfterDecimal</i>	Optional. Numeric value indicating how many places to the right of the decimal are displayed. Default value is -1, which indicates that the computer's regional settings are used.
<i>IncludeLeadingDigit</i>	Optional. Tristate constant that indicates whether or not a leading zero is displayed for fractional values. See Settings section for values.
<i>UseParensForNegativeNumbers</i>	Optional. Tristate constant that indicates whether or not to place negative values within parentheses. See Settings section for values.
<i>GroupDigits</i>	Optional. Tristate constant that indicates whether or not numbers are grouped using the group delimiter specified in the computer's regional settings. See Settings section for values.

Settings

The *IncludeLeadingDigit*, *UseParensForNegativeNumbers*, and *GroupDigits*

arguments have the following settings:

Constant	Value	Description
vbTrue	-1	True
vbFalse	0	False
vbUseDefault	-2	Use the setting from the computer's regional settings.

Remarks

When one or more optional arguments are omitted, the values for omitted arguments are provided by the computer's regional settings.

Note All settings information comes from the **Regional Settings Number** tab.

[Show All](#)

GUIDFromString Function

The **GUIDFromString** function converts a [string](#) to a [GUID](#), which is an [array](#) of type [Byte](#).

GUIDFromString(*stringexpression*)

The **GUIDFromString** function has the following argument.

Argument	Description
<i>stringexpression</i>	A string expression which evaluates to a GUID in string form.

Remarks

The [Microsoft Jet database engine](#) stores GUIDs as arrays of type **Byte**. However, Microsoft Access can't return **Byte** data from a [control](#) on a [form](#) or [report](#). In order to return the value of a GUID from a control, you must convert it to a string. To convert a GUID to a string, use the **StringFromGUID** function. To convert a string to a GUID, use the **GUIDFromString** function.

Hex Function

Returns a String representing the hexadecimal value of a number.

Syntax

Hex(*number*)

The required *number* argument is any valid numeric expression or string expression .

Remarks

If *number* is not already a whole number, it is rounded to the nearest whole number before being evaluated.

If <i>number</i> is	Hex returns
Null	Null
Empty	Zero (0)
Any other number	Up to eight hexadecimal characters

You can represent hexadecimal numbers directly by preceding numbers in the proper range with &H. For example, &H10 represents decimal 16 in hexadecimal notation.

Hour Function

Returns a **Variant (Integer)** specifying a whole number between 0 and 23, inclusive, representing the hour of the day.

Syntax

Hour(*time*)

The required *time* argument is any Variant, numeric expression, string expression, or any combination, that can represent a time. If *time* contains Null, **Null** is returned.

Minute Function

Returns a **Variant (Integer)** specifying a whole number between 0 and 59, inclusive, representing the minute of the hour.

Syntax

Minute(*time*)

The required *time* argument is any Variant, numeric expression, string expression, or any combination, that can represent a time. If *time* contains Null, **Null** is returned.

Month Function

Returns a **Variant (Integer)** specifying a whole number between 1 and 12, inclusive, representing the month of the year.

Syntax

Month(*date*)

The required *date* argument is any Variant, numeric expression, string expression, or any combination, that can represent a date. If *date* contains Null, **Null** is returned.

Note If the **Calendar** property setting is Gregorian, the returned integer represents the Gregorian day of the week for the date argument. If the calendar is Hijri, the returned integer represents the Hijri day of the week for the date argument. For Hijri dates, the argument number is any numeric expression that can represent a date and/or time from 1/1/100 (Gregorian Aug 2, 718) through 4/3/9666 (Gregorian Dec 31, 9999).

[Show All](#)

Nz Function

You can use the **Nz** function to return zero, a [zero-length string](#) (" "), or another specified value when a **Variant** is **Null**. For example, you can use this function to convert a **Null** value to another value and prevent it from propagating through an expression.

Nz(*variant*, [*valueifnull*])

The **Nz** function has the following arguments.

Argument	Description
<i>variant</i>	A variable of data type Variant . Optional (unless used in a query). A Variant that supplies a value to be returned if the <i>variant</i> argument is Null . This argument enables you to return a value other than zero or a zero-length string.
<i>valueifnull</i>	Note If you use the Nz function in an expression in a query without using the <i>valueifnull</i> argument, the results will be a zero-length string in the fields that contain null values.

If the value of the *variant* argument is **Null**, the **Nz** function returns the number zero or a zero-length string (always returns a zero-length string when used in a query expression), depending on whether the context indicates the value should be a number or a string. If the optional *valueifnull* argument is included, then the **Nz** function will return the value specified by that argument if the *variant* argument is **Null**. When used in a query expression, the **NZ** function should always include the *valueifnull* argument,

If the value of *variant* isn't **Null**, then the **Nz** function returns the value of *variant*.

Remarks

The **Nz** function is useful for expressions that may include **Null** values. To force an expression to evaluate to a non-**Null** value even when it contains a **Null** value, use the **Nz** function to return zero, a zero-length string, or a custom return value.

For example, the expression `2 + varX` will always return a **Null** value when the **Variant** `varX` is **Null**. However, `2 + Nz(varX)` returns 2.

You can often use the **Nz** function as an alternative to the **IIf** function. For example, in the following code, two expressions including the **IIf** function are necessary to return the desired result. The first expression including the **IIf** function is used to check the value of a variable and convert it to zero if it is **Null**.

```
varTemp = IIf(IsNull(varFreight), 0, varFreight)
varResult = IIf(varTemp > 50, "High", "Low")
```

In the next example, the **Nz** function provides the same functionality as the first expression, and the desired result is achieved in one step rather than two.

```
varResult = IIf(Nz(varFreight) > 50, "High", "Low")
```

If you supply a value for the optional argument *valueifnull*, that value will be returned when *variant* is **Null**. By including this optional argument, you may be able to avoid the use of an expression containing the **IIf** function. For example, the following expression uses the **IIf** function to return a string if the value of `varFreight` is **Null**.

```
varResult = IIf(IsNull(varFreight), "No Freight Charge", varFreight)
```

In the next example, the optional argument supplied to the **Nz** function provides the string to be returned if `varFreight` is **Null**.

```
varResult = Nz(varFreight, "No Freight Charge")
```

Oct Function

Returns a **Variant (String)** representing the octal value of a number.

Syntax

Oct(*number*)

The required *number* argument is any valid numeric expression or string expression.

Remarks

If *number* is not already a whole number, it is rounded to the nearest whole number before being evaluated.

If <i>number</i> is	Oct returns
Null	Null
Empty	Zero (0)
Any other number	Up to 11 octal characters

You can represent octal numbers directly by preceding numbers in the proper range with &0. For example, &010 is the octal notation for decimal 8.

Second Function

Returns a **Variant (Integer)** specifying a whole number between 0 and 59, inclusive, representing the second of the minute.

Syntax

Second(*time*)

The required *time* argument is any Variant, numeric expression, string expression, or any combination, that can represent a time. If *time* contains Null, **Null** is returned.

Str Function

Returns a **Variant (String)** representation of a number.

Syntax

Str(*number*)

The required *number* argument is a Long containing any valid numeric expression.

Remarks

When numbers are converted to strings, a leading space is always reserved for the sign of *number*. If *number* is positive, the returned string contains a leading space and the plus sign is implied.

Use the **Format** function to convert numeric values you want formatted as dates, times, or currency or in other user-defined formats. Unlike **Str**, the **Format** function doesn't include a leading space for the sign of *number*.

Note The **Str** function recognizes only the period (.) as a valid decimal separator. When different decimal separators may be used (for example, in international applications), use **CStr** to convert a number to a string.

StrConv Function

Returns a **Variant (String)** converted as specified.

Syntax

StrConv(*string*, *conversion*, *LCID*)

The **StrConv** function syntax has these named arguments:

Part	Description
<i>string</i>	Required. String expression to be converted.
<i>conversion</i>	Required. Integer. The sum of values specifying the type of conversion to perform.
<i>LCID</i>	Optional. The LocaleID, if different than the system LocaleID. (The system LocaleID is the default.)

Settings

The *conversion* argument settings are:

Constant	Value	Description
vbUpperCase	1	Converts the string to uppercase characters.
vbLowerCase	2	Converts the string to lowercase characters.
vbProperCase	3	Converts the first letter of every word in string to uppercase.
vbWide*	4*	Converts narrow (single-byte) characters in string to wide (double-byte) characters.
vbNarrow*	8*	Converts wide (double-byte) characters in string to narrow (single-byte) characters.
vbKatakana**	16**	Converts Hiragana characters in string to Katakana characters.
vbHiragana**	32**	Converts Katakana characters in string to Hiragana characters.

vbUnicode	64	Converts the string to Unicode using the default code page of the system. (Not available on the Macintosh.)
vbFromUnicode	128	Converts the string from Unicode to the default code page of the system. (Not available on the Macintosh.)

*Applies to Far East locales.

**Applies to Japan only.

Note These constants are specified by Visual Basic for Applications. As a result, they may be used anywhere in your code in place of the actual values. Most can be combined, for example, **vbUpperCase** + **vbWide**, except when they are mutually exclusive, for example, **vbUnicode** + **vbFromUnicode**. The constants **vbWide**, **vbNarrow**, **vbKatakana**, and **vbHiragana** cause run-time errors when used in locales where they do not apply.

The following are valid word separators for proper casing: Null (**Chr\$(0)**), horizontal tab (**Chr\$(9)**), linefeed (**Chr\$(10)**), vertical tab (**Chr\$(11)**), form feed (**Chr\$(12)**), carriage return (**Chr\$(13)**), space (SBCS) (**Chr\$(32)**). The actual value for a space varies by country for DBCS.

Remarks

When you're converting from a **Byte** array in ANSI format to a string, you should use the **StrConv** function. When you're converting from such an array in Unicode format, use an assignment statement.

TimeSerial Function

Returns a **Variant (Date)** containing the time for a specific hour, minute, and second.

Syntax

TimeSerial(*hour, minute, second*)

The **TimeSerial** function syntax has these named arguments:

Part	Description
<i>hour</i>	Required; Variant (Integer) . Number between 0 (12:00 A.M.) and 23 (11:00 P.M.), inclusive, or a numeric expression.
<i>minute</i>	Required; Variant (Integer) . Any numeric expression.
<i>second</i>	Required; Variant (Integer) . Any numeric expression.

Remarks

To specify a time, such as 11:59:59, the range of numbers for each **TimeSerial** argument should be in the normal range for the unit; that is, 0–23 for hours and 0–59 for minutes and seconds. However, you can also specify relative times for each argument using any numeric expression that represents some number of hours, minutes, or seconds before or after a certain time. The following example uses expressions instead of absolute time numbers. The **TimeSerial** function returns a time for 15 minutes before (-15) six hours before noon (12 - 6), or 5:45:00 A.M.

```
TimeSerial(12 - 6, -15, 0)
```

When any argument exceeds the normal range for that argument, it increments to the next larger unit as appropriate. For example, if you specify 75 minutes, it is evaluated as one hour and 15 minutes. If any single argument is outside the range -32,768 to 32,767, an error occurs. If the time specified by the three arguments causes the date to fall outside the acceptable range of dates, an error

OCCURS.

TimeValue Function

Returns a **Variant (Date)** containing the time.

Syntax

TimeValue(*time*)

The required *time* argument is normally a string expression representing a time from 0:00:00 (12:00:00 A.M.) to 23:59:59 (11:59:59 P.M.), inclusive. However, *time* can also be any expression that represents a time in that range. If *time* contains Null, **Null** is returned.

Remarks

You can enter valid times using a 12-hour or 24-hour clock. For example, "2:24PM" and "14:24" are both valid *time* arguments.

If the *time* argument contains date information, **TimeValue** doesn't return it. However, if *time* includes invalid date information, an error occurs.

Type Conversion Functions

Each function coerces an expression to a specific data type.

Syntax

CBool(*expression*)

CByte(*expression*)

CCur(*expression*)

CDate(*expression*)

CDBl(*expression*)

CDec(*expression*)

CInt(*expression*)

CLng(*expression*)

CSng(*expression*)

CStr(*expression*)

CVar(*expression*)

The required *expression* argument is any string expression or numeric expression.

Return Types

The function name determines the return type as shown in the following:

Function	Return Type	Range for <i>expression</i> argument
-----------------	--------------------	---

CBool	Boolean	Any valid string or numeric expression.
CByte	Byte	0 to 255.
CCur	Currency	-922,337,203,685,477.5808 to 922,337,203,685,477.5807.
CDate	Date	Any valid date expression.
CDbl	Double	-1.79769313486231E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values. +/-79,228,162,514,264,337,593,543,950,335 for zero-scaled numbers, that is, numbers with no decimal places. For numbers with 28 decimal places, the range is +/-7.9228162514264337593543950335. The smallest possible non-zero number is 0.000000000000000000000000000001.
CDec	Decimal	
CInt	Integer	-32,768 to 32,767; fractions are rounded.
CLng	Long	-2,147,483,648 to 2,147,483,647; fractions are rounded.
CSng	Single	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
CStr	String	Returns for CStr depend on the <i>expression</i> argument.
CVar	Variant	Same range as Double for numerics. Same range as String for non-numerics.

Remarks

If the *expression* passed to the function is outside the range of the data type being converted to, an error occurs.

In general, you can document your code using the data-type conversion functions to show that the result of some operation should be expressed as a particular data type rather than the default data type. For example, use **CCur** to force currency arithmetic in cases where single-precision, double-precision, or integer arithmetic normally would occur.

You should use the data-type conversion functions instead of **Val** to provide internationally aware conversions from one data type to another. For example, when you use **CCur**, different decimal separators, different thousand separators,

and various currency options are properly recognized depending on the locale setting of your computer.

When the fractional part is exactly 0.5, **CInt** and **CLng** always round it to the nearest even number. For example, 0.5 rounds to 0, and 1.5 rounds to 2. **CInt** and **CLng** differ from the **Fix** and **Int** functions, which truncate, rather than round, the fractional part of a number. Also, **Fix** and **Int** always return a value of the same type as is passed in.

Use the **IsDate** function to determine if *date* can be converted to a date or time. **CDate** recognizes date literals and time literals as well as some numbers that fall within the range of acceptable dates. When converting a number to a date, the whole number portion is converted to a date. Any fractional part of the number is converted to a time of day, starting at midnight.

CDate recognizes date formats according to the locale setting of your system. The correct order of day, month, and year may not be determined if it is provided in a format other than one of the recognized date settings. In addition, a long date format is not recognized if it also contains the day-of-the-week string.

A **CVDate** function is also provided for compatibility with previous versions of Visual Basic. The syntax of the **CVDate** function is identical to the **CDate** function, however, **CVDate** returns a **Variant** whose subtype is **Date** instead of an actual **Date** type. Since there is now an intrinsic **Date** type, there is no further need for **CVDate**. The same effect can be achieved by converting an expression to a **Date**, and then assigning it to a **Variant**. This technique is consistent with the conversion of all other intrinsic types to their equivalent **Variant** subtypes.

Note The **CDec** function does not return a discrete data type; instead, it always returns a **Variant** whose value has been converted to a **Decimal** subtype.

Val Function

Returns the numbers contained in a string as a numeric value of appropriate type.

Syntax

Val(*string*)

The required *string* argument is any valid string expression.

Remarks

The **Val** function stops reading the string at the first character it can't recognize as part of a number. Symbols and characters that are often considered parts of numeric values, such as dollar signs and commas, are not recognized. However, the function recognizes the radix prefixes &O (for octal) and &H (for hexadecimal). Blanks, tabs, and linefeed characters are stripped from the argument.

The following returns the value 1615198:

```
Val("    1615 198th Street N.E.")
```

In the code below, **Val** returns the decimal value -1 for the hexadecimal value shown:

```
Val("&HFFFF")
```

Note The **Val** function recognizes only the period (.) as a valid decimal separator. When different decimal separators are used, as in international applications, use **Cdbl** instead to convert a string to a number.

Weekday Function

Returns a **Variant (Integer)** containing a whole number representing the day of the week.

Syntax

Weekday(*date*, [*firstdayofweek*])

The **Weekday** function syntax has these named arguments:

Part	Description
<i>date</i>	Required. Variant, numeric expression, string expression, or any combination, that can represent a date. If <i>date</i> contains Null, Null is returned.
<i>firstdayofweek</i>	Optional. A constant that specifies the first day of the week. If not specified, vbSunday is assumed.

Settings

The *firstdayofweek* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use the NLS API setting.
vbSunday	1	Sunday (default)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

Return Values

The **Weekday** function can return any of these values:

Constant	Value	Description
vbSunday	1	Sunday
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

Remarks

If the **Calendar** property setting is Gregorian, the returned integer represents the Gregorian day of the week for the date argument. If the calendar is Hijri, the returned integer represents the Hijri day of the week for the date argument. For Hijri dates, the argument number is any numeric expression that can represent a date and/or time from 1/1/100 (Gregorian Aug 2, 718) through 4/3/9666 (Gregorian Dec 31, 9999).

Year Function

Returns a **Variant (Integer)** containing a whole number representing the year.

Syntax

Year(*date*)

The required *date* argument is any Variant, numeric expression, string expression, or any combination, that can represent a date. If *date* contains Null, **Null** is returned.

Note If the **Calendar** property setting is Gregorian, the returned integer represents the Gregorian year for the date argument. If the calendar is Hijri, the returned integer represents the Hijri year for the date argument. For Hijri dates, the argument number is any numeric expression that can represent a date and/or time from 1/1/100 (Gregorian Aug 2, 718) through 4/3/9666 (Gregorian Dec 31, 9999).

[Show All](#)

CodeDb Method

You can use the **CodeDb** method in a code module to determine the name of the **Database** object that refers to the database in which code is currently running. Use the **CodeDb** method to access [Data Access Objects \(DAO\)](#) that are part of a [library database](#).

expression.**CodeDb**

expression Required. An expression that returns one of the objects in the Applies To list.

Remarks

For example, you can use the **CodeDb** method in a module in a library database to create a **Database** object referring to the library database. You can then open a [recordset](#) based on a table in the library database.

Set database = CodeDb

The **CodeDb** method returns a **Database** object for which the **Name** property is the full path and name of the database from which it is called. This method can be useful when you need to manipulate the Data Access Objects in your library database.

When you call a method in a library database, the database from which you have called the method remains the current database, even while code is running in a module in the library database. In order to refer to the Data Access Objects in the library database, you need to know the name of the **Database** object that represents the library database.

For example, suppose you have a table in a library database that lists error messages. To manipulate data in the table from code, you could use the **CodeDb** method to determine the name of the **Database** object that refers to the library database that contains the table.

If the **CodeDb** method is run from the current database, it returns the name of the current database, which is the same value returned by the **CurrentDb** method.

[Show All](#)

CreateControl Method

The **CreateControl** method creates a [control](#) on a specified open form. For example, suppose you are building a custom wizard that allows users to easily construct a particular form. You can use the **CreateControl** method in your wizard to add the appropriate controls to the form.

CreateControl(*formname*, *controltype*[, *section*[, *parent*[, *columnname*[, *left*[, *top*[, *width*[, *height*]]]]]])

The **CreateControl** method has the following arguments.

Argument	Description
<i>formname</i>	A string expression identifying the name of the open form or report on which you want to create the control.
<i>controltype</i>	One of the following intrinsic constants identifying the type of control you want to create. To view these constants and paste them into your code from the Object Browser, click Object Browser on the Visual Basic toolbar , then click Access in the Project/Library box, and click AcControlType in the Classes box.

Constant	Control
acBoundObjectFrame	Bound object frame
acCheckBox	Check box
acComboBox	Combo box
acCommandButton	Command button
acCustomControl	ActiveX control
acImage	Image

acLabel	Label
acLine	Line
acListBox	List box
acObjectFrame	Unbound object frame
acOptionButton	Option button
acOptionGroup	Option group
acPage	Page
acPageBreak	Page break
acRectangle	Rectangle
acSubform	Subform
acTabCtl	Tab control
acTextBox	Text box
acToggleButton	Toggle button

section

One of the following intrinsic constants identifying the section that will contain the new control. To view these constants and paste them into your code from the Object Browser, click **Object Browser** on the **Visual Basic toolbar**, then click **Access** in the **Project/Library** box, and click **AcSection** in the **Classes** box.

Constant	Section
acDetail	(Default) Detail section
acHeader	Form or report header
acFooter	Form or report footer

acPageHeader	Page header
acPageFooter	Page footer
acGroupLevel1Header	Group-level 1 header (reports only)
acGroupLevel1Footer	Group-level 1 footer (reports only)
acGroupLevel2Header	Group-level 2 header (reports only)
acGroupLevel2Footer	Group-level 2 footer (reports only)

If a report has additional group levels, the header/footer pairs are numbered consecutively, beginning with 9.

parent A string expression identifying the name of the parent control of an attached control. For controls that have no parent control, use a [zero-length string](#) for this argument, or omit it.

columnname The name of the field to which the control will be bound, if it is to be a data-bound control.

If you are creating a control that won't be bound to a field, use a zero-length string for this argument.

left, top [Numeric expressions](#) indicating the coordinates for the upper-left corner of the control in [twips](#).

width,
height

Numeric expressions indicating the width and height of the control in twips.

Remarks

You can use the **CreateControl** and **CreateReportControl** methods in a custom wizard to create controls on a form or report. Both methods return a **Control** object.

You can use the **CreateControl** and **CreateReportControl** methods only in [form Design view](#) or [report Design view](#), respectively.

You use the *parent* argument to identify the relationship between a main control and a subordinate control. For example, if a text box has an attached label, the text box is the main (or parent) control and the label is the subordinate (or child) control. When you create the label control, set its *parent* argument to a string identifying the name of the parent control. When you create the text box, set its *parent* argument to a zero-length string.

You also set the *parent* argument when you create check boxes, option buttons, or toggle buttons. An option group is the parent control of any check boxes, option buttons, or toggle buttons that it contains. The only controls that can have a parent control are a label, check box, option button, or toggle button. All of these controls can also be created independently, without a parent control.

Set the *columnname* argument according to the type of control you are creating and whether or not it will be [bound](#) to a field in a table. The controls that may be bound to a field include the text box, list box, combo box, option group, and bound object frame. Additionally, the toggle button, option button, and check box controls may be bound to a field if they are not contained in an option group.

If you specify the name of a field for the *columnname* argument, you create a control that is bound to that field. All of the control's properties are then automatically set to the settings of any corresponding field properties. For example, the value of the control's **ValidationRule** property will be the same as the value of that property for the field.

Note If your wizard creates controls on a new or existing form or report, it must first open the form or report in [Design view](#).

To remove a control from a form or report, use the **DeleteControl** and **DeleteReportControl** statements.

[Show All](#)

CreateForm Method

The **CreateForm** method creates a form and returns a **Form** object.

CreateForm([*database*[, *formtemplate*]])

The **CreateForm** method has the following arguments.

Argument	Description
<i>database</i>	A string expression identifying the name of the database that contains the form template you want to use to create a form. If you want the current database, omit this argument. If you want to use an open library database , specify the library database with this argument.
<i>formtemplate</i>	A string expression identifying the name of the form you want to use as a template to create a new form. If you omit this argument, Microsoft Access bases the new form on the template specified by the Forms/Reports tab of the Options dialog box, available by clicking Options on the Tools menu.

Remarks

You can use the **CreateForm** method when designing a wizard that creates a new form.

The **CreateForm** method opens a new, minimized form in [form Design view](#).

If the name you use for the *formtemplate* argument isn't valid, Visual Basic uses the form template specified by the **Form Template** setting on the **Forms/Reports** tab of the **Options** dialog box.

The **CreateForm** method creates minimized forms and reports.

[Show All](#)

CreateGroupLevel Method

You can use the **CreateGroupLevel** method to specify a field or [expression](#) on which to group or sort data in a report. **Long.**

expression.**CreateGroupLevel**(*ReportName*, *Expression*, *Header*, *Footer*)

expression Required. An expression that returns one of the objects in the Applies To list.

ReportName Required **String**. A [string expression](#) identifying the name of the report that will contain the new group level.

Expression Required **String**. A string expression identifying the field or expression to sort or group on.

Header Required **Integer**. An [Integer](#) value that indicates a field or expression will have an associated group header. If the *header* argument is **True** (-1), the field or expression will have a group header. If the *header* argument is **False** (0), the field or expression won't. You can create a header by setting the argument to **True**.

Footer Required **Integer**. An [Integer](#) value that indicates a field or expression will have an associated group footer. If the *footer* argument is **True** (-1), the field or expression will have a group footer. If the *footer* argument is **False** (0), the field or expression won't. You can create a footer by setting the argument to **True**.

Remarks

For example, suppose you are building a custom wizard that provides the user with a choice of fields on which to group data when designing a report. Call the **CreateGroupLevel** method from your wizard to create the appropriate groups according to the user's choice.

You can use the **CreateGroupLevel** method when designing a wizard that creates a report with groups or totals. The **CreateGroupLevel** method groups or sorts data on the specified field or expression and creates a header and/or footer for the group level.

The **CreateGroupLevel** method is available only in [report Design view](#).

Microsoft Access uses an [array](#), the **GroupLevel** property array, to keep track of the group levels created for a report. The **CreateGroupLevel** method adds a new group level to the array, based on the *expression* argument. The **CreateGroupLevel** method then returns an index value that represents the new group level's position in the array. The first field or expression you sort or group on is level 0, the second is level 1, and so on. You can have up to ten group levels in a report (0 to 9).

When you specify that either the *header* or *footer* argument, or both, is **True**, the **GroupHeader** and **GroupFooter** properties in a report are set to Yes, and a header and/or footer is created for the group level.

Once a header or footer is created, you can set other GroupLevel properties: **GroupOn** , **GroupInterval** , and **KeepTogether** . You can set these properties in Visual Basic or in the report's **Sorting And Grouping** box, available by clicking **Sorting And Grouping**  on the **Report Design toolbar**.

Note If your wizard creates group levels in a new or existing report, it must open the report in [Design view](#).

CreateObject Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Creates and returns a reference to an ActiveX object.

Syntax

CreateObject(*class*,[*servername*])

The **CreateObject** function syntax has these parts:

Part	Description
<i>class</i>	Required; Variant (String) . The application name and class of the object to create.
<i>servername</i>	Optional; Variant (String) . The name of the network server where the object will be created. If <i>servername</i> is an empty string (""), the local machine is used.

The *class* argument uses the syntax *appname.objecttype* and has these parts:

Part	Description
<i>appname</i>	Required; Variant (String) . The name of the application providing the object.
<i>objecttype</i>	Required; Variant (String) . The type or class of object to create.

Remarks

Every application that supports Automation provides at least one type of object. For example, a word processing application may provide an **Application** object, a **Document** object, and a **Toolbar** object.

To create an ActiveX object, assign the object returned by **CreateObject** to an object variable:

```
' Declare an object variable to hold the object  
' reference. Dim as Object causes late binding.  
Dim ExcelSheet As Object  
Set ExcelSheet = CreateObject("Excel.Sheet")
```

This code starts the application creating the object, in this case, a Microsoft Excel spreadsheet. Once an object is created, you reference it in code using the object variable you defined. In the following example, you access properties and methods of the new object using the object variable, `ExcelSheet`, and other Microsoft Excel objects, including the `Application` object and the `Cells` collection.

```
' Make Excel visible through the Application object.  
ExcelSheet.Application.Visible = True  
' Place some text in the first cell of the sheet.  
ExcelSheet.Application.Cells(1, 1).Value = "This is column A, row 1"  
' Save the sheet to C:\test.xls directory.  
ExcelSheet.SaveAs "C:\TEST.XLS"  
' Close Excel with the Quit method on the Application object.  
ExcelSheet.Application.Quit  
' Release the object variable.  
Set ExcelSheet = Nothing
```

Declaring an object variable with the `As Object` clause creates a variable that can contain a reference to any type of object. However, access to the object through that variable is late bound; that is, the binding occurs when your program is run. To create an object variable that results in early binding, that is, binding when the program is compiled, declare the object variable with a specific class ID. For example, you can declare and create the following Microsoft Excel references:

```
Dim xlApp As Excel.Application  
Dim xlBook As Excel.Workbook  
Dim xlSheet As Excel.WorkSheet  
Set xlApp = CreateObject("Excel.Application")  
Set xlBook = xlApp.Workbooks.Add  
Set xlSheet = xlBook.Worksheets(1)
```

The reference through an early-bound variable can give better performance, but can only contain a reference to the class specified in the declaration.

You can pass an object returned by the **CreateObject** function to a function expecting an object as an argument. For example, the following code creates and passes a reference to a Excel.Application object:

```
Call MySub (CreateObject("Excel.Application"))
```

You can create an object on a remote networked computer by passing the name of the computer to the *servername* argument of **CreateObject**. That name is the same as the Machine Name portion of a share name: for a share named "\\MyServer\Public," *servername* is "MyServer."

Note Refer to COM documentation (see *Microsoft Developer Network*) for additional information on making an application visible on a remote networked computer. You may have to add a registry key for your application.

The following code returns the version number of an instance of Excel running on a remote computer named MyServer:

```
Dim xlApp As Object  
Set xlApp = CreateObject("Excel.Application", "MyServer")  
Debug.Print xlApp.Version
```

If the remote server doesn't exist or is unavailable, a run-time error occurs.

Note Use **CreateObject** when there is no current instance of the object. If an instance of the object is already running, a new instance is started, and an object of the specified type is created. To use the current instance, or to start the application and have it load a file, use the **GetObject** function.

If an object has registered itself as a single-instance object, only one instance of the object is created, no matter how many times **CreateObject** is executed.

[Show All](#)

CreateReport Method

The **CreateReport** method creates a report and returns a **Report** object. For example, suppose you are building a custom wizard to create a sales report. You can use the **CreateReport** method in your wizard to create a new report based on a specified report template.

expression.**CreateReport**(*Database*, *ReportTemplate*)

expression Required. An expression that returns one of the objects in the Applies To list.

Database Optional **Variant**. A [string expression](#) identifying the name of the database that contains the report template you want to use to create a report. If you want the current database, omit this argument. If you want to use an open [library database](#), specify the library database with this argument.

ReportTemplate Optional **Variant**. A string expression identifying the name of the report you want to use as a template to create a new report. If you omit this argument, Microsoft Access bases the new report on the template specified by the **Forms/Reports** tab of the **Options** dialog box, available by clicking **Options** on the **Tools** menu.

Remarks

You can use the **CreateReport** method when designing a wizard that creates a new report.

The **CreateReport** method open a new, minimized report in [report Design view](#).

If the name you use for the *reporttemplate* argument isn't valid, Visual Basic uses the report template specified by the **Report Template** setting on the **Forms/Reports** tab of the **Options** dialog box.

The **CreateReport** method creates minimized forms and reports.

[Show All](#)

CreateReportControl Method

The **CreateReportControl** method creates a control on a specified open report. For more information, see the **CreateControl** method.

expression.**CreateReportControl**(*ReportName*, *ControlType*, *Section*, *Parent*, *ColumnName*, *Left*, *Top*, *Width*, *Height*)

expression Required. An expression that returns one of the objects in the Applies To list.

ReportName Required **String**. A [string expression](#) identifying the name of the open report on which you want to create the control.

ControlType Required **AcControlType** . The type of control you want to create.

AcControlType can be one of these AcControlType constants.

acBoundObjectFrame

acCheckBox

acComboBox

acCommandButton

acCustomControl

acImage

acLabel

acLine

acListBox

acObjectFrame

acOptionButton

acOptionGroup

acPage

acPageBreak

acRectangle

acSubform

acTabCtl
acTextBox
acToggleButton

Section Optional **AcSection** . The section that will contain the new control.

AcSection can be one of these AcSection constants.

acDetail *default*
acFooter
acGroupLevel1Footer
acGroupLevel1Header
acGroupLevel2Footer
acGroupLevel2Header
acHeader
acPageFooter
acPageHeader

Parent Optional **Variant**. A string expression identifying the name of the parent control of an attached control. For controls that have no parent control, use a [zero-length string](#) for this argument, or omit it.

ColumnName Optional **Variant**. The name of the field to which the control will be bound, if it is to be a data-bound control.

Left, Top Optional **Variant**. [Numeric expressions](#) indicating the coordinates for the upper-left corner of the control in [twips](#).

Width, Height Optional **Variant**. Numeric expressions indicating the width and height of the control in twips.

[Show All](#)

CurrentDb Method

The **CurrentDb** method returns an object variable of type **Database** that represents the database currently open in the Microsoft Access window.

expression.**CurrentDb**

expression Required. An expression that returns one of the objects in the Applies To list.

Remarks

Note In Microsoft Access the **CurrentDb** method establishes a hidden reference to the Microsoft DAO 3.6 Object Library in a [Microsoft Access database](#) (.mdb).

In order to manipulate the structure of your database and its data from Visual Basic, you must use [Data Access Objects \(DAO\)](#). The **CurrentDb** method provides a way to access the current database from Visual Basic code without having to know the name of the database. Once you have a variable that points to the current database, you can also access and manipulate other objects and collections in the DAO hierarchy .

You can use the **CurrentDb** method to create multiple object variables that refer to the current database. In the following example, the variables dbsA and dbsB both refer to the current database:

```
Dim dbsA As Database, dbsB As Database
Set dbsA = CurrentDb
Set dbsB = CurrentDb
```

Note In previous versions of Microsoft Access, you may have used the syntax `DBEngine.Workspaces(0).Databases(0)` or `DBEngine(0)(0)` to return a pointer to the current database. In Microsoft Access 2000, you should use the **CurrentDb** method instead. The **CurrentDb** method creates another instance of the current database, while the `DBEngine(0)(0)` syntax refers to the open copy of the current database. The **CurrentDb** method enables you to create more than one variable of type **Database** that refers to the current database. Microsoft Access still supports the `DBEngine(0)(0)` syntax, but you should consider making this modification to your code in order to avoid possible conflicts in a [multiuser database](#).

If you need to work with another database at the same time that the current database is open in the Microsoft Access window, use the **OpenDatabase** method of a **Workspace** object. The **OpenDatabase** method doesn't actually open the second database in the Microsoft Access window; it simply returns a **Database** variable representing the second database. The following example returns a pointer to the current database and to a database called Contacts.mdb:

```
Dim dbsCurrent As Database, dbsContacts As Database
Set dbsCurrent = CurrentDb
Set dbsContacts = DBEngine.Workspaces(0).OpenDatabase("Contacts.mdb")
```

[Show All](#)

CurrentUser Method

You can use the **CurrentUser** method to return the name of the current user of the database. **String**.

expression.**CurrentUser**

expression Required. An expression that returns one of the objects in the Applies To list.

Remarks

For example, use the **CurrentUser** method in a procedure that keeps track of the users who modify the database.

The **CurrentUser** method returns a [string](#) that contains the name of the current [user account](#).

If you haven't established a [security-enabled workgroup](#), the **CurrentUser** method returns the name of the default user account, Admin. The Admin user account gives the user full permissions to all [database objects](#).

If you have enabled workgroup security, then the **CurrentUser** method returns the name of the current user account. For user accounts other than Admin, you can specify [permissions](#) that restrict the users' access to database objects.

[Show All](#)

DeleteControl Method

The **DeleteControl** method deletes a specified [control](#) from a form.

expression.DeleteControl(FormName, ControlName)

expression Required. An expression that returns one of the objects in the Applies To list.

FormName Required **String**. A [string expression](#) identifying the name of the form or report containing the control you want to delete.

ControlName Required **String**. A string expression identifying the name of the control you want to delete.

Remarks

For example, suppose you have a procedure that must be run the first time each user logs onto your database. You can set the **OnClick** property of a button on the form to this procedure. Once the user has logged on and run the procedure, you can use the **DeleteControl** method to dynamically remove the [command button](#) from the form.

The **DeleteControl** method is available only in [form Design view](#) or [report Design view](#), respectively.

Note If you are building a wizard that deletes a control from a form or report, your wizard must open the form or report in [Design view](#) before it can delete the control.

[Show All](#)

DeleteReportControl Method

The **DeleteReportControl** method deletes a specified control from a report.

expression.DeleteReportControl(ReportName, ControlName)

expression Required. An expression that returns one of the objects in the Applies To list.

ReportName Required **String**. A [string expression](#) identifying the name of the form or report containing the control you want to delete.

ControlName Required **String**. A string expression identifying the name of the control you want to delete.

Remarks

The **DeleteReportControl** method is available only in [form Design view](#) or [report Design view](#), respectively.

Note If you are building a wizard that deletes a control from a form or report, your wizard must open the form or report in [Design view](#) before it can delete the control.

GetObject Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a reference to an object provided by an ActiveX component.

Syntax

GetObject([*pathname*] [, *class*])

The **GetObject** function syntax has these named arguments:

Part	Description
<i>pathname</i>	Optional; Variant (String) . The full path and name of the file containing the object to retrieve. If <i>pathname</i> is omitted, <i>class</i> is required.
<i>class</i>	Optional; Variant (String) . A string representing the class of the object.

The *class* argument uses the syntax *appname.objecttype* and has these parts:

Part	Description
<i>appname</i>	Required; Variant (String) . The name of the application providing the object.
<i>objecttype</i>	Required; Variant (String) . The type or class of object to create.

Remarks

Use the **GetObject** function to access an ActiveX object from a file and assign the object to an object variable. Use the **Set** statement to assign the object returned by **GetObject** to the object variable. For example:

```
Dim CADObject As Object
Set CADObject = GetObject("C:\CAD\SCHEMA.CAD")
```

When this code is executed, the application associated with the specified *pathname* is started and the object in the specified file is activated.

If *pathname* is a zero-length string (""), **GetObject** returns a new object instance of the specified type. If the *pathname* argument is omitted, **GetObject** returns a currently active object of the specified type. If no object of the specified type exists, an error occurs.

Some applications allow you to activate part of a file. Add an exclamation point (!) to the end of the file name and follow it with a string that identifies the part of the file you want to activate. For information on how to create this string, see the documentation for the application that created the object.

For example, in a drawing application you might have multiple layers to a drawing stored in a file. You could use the following code to activate a layer within a drawing called SCHEMA.CAD:

```
Set LayerObject = GetObject("C:\CAD\SCHEMA.CAD!Layer3")
```

If you don't specify the object's *class*, Automation determines the application to start and the object to activate, based on the file name you provide. Some files, however, may support more than one class of object. For example, a drawing might support three different types of objects: an **Application** object, a **Drawing** object, and a **Toolbar** object, all of which are part of the same file. To specify which object in a file you want to activate, use the optional *class* argument. For example:

```
Dim MyObject As Object
Set MyObject = GetObject("C:\DRAWINGS\SAMPLE.DRW", "FIGMENT.DRAWING")
```

In the example, FIGMENT is the name of a drawing application and DRAWING is one of the object types it supports.

Once an object is activated, you reference it in code using the object variable you defined. In the preceding example, you access properties and methods of the new object using the object variable MyObject. For example:

```
MyObject.Line 9, 90  
MyObject.InsertText 9, 100, "Hello, world."  
MyObject.SaveAs "C:\DRAWINGS\SAMPLE.DRW"
```

Note Use the **GetObject** function when there is a current instance of the object or if you want to create the object with a file already loaded. If there is no current instance, and you don't want the object started with a file loaded, use the **CreateObject** function.

If an object has registered itself as a single-instance object, only one instance of the object is created, no matter how many times **CreateObject** is executed. With a single-instance object, **GetObject** always returns the same instance when called with the zero-length string ("") syntax, and it causes an error if the *pathname* argument is omitted. You can't use **GetObject** to obtain a reference to a class created with Visual Basic.

[Show All](#)

HyperlinkPart Method

The **HyperlinkPart** method returns information about data stored as a [Hyperlink data type](#). **String**.

expression.**HyperlinkPart**(*Hyperlink*, *Part*)

expression Required. An expression that returns one of the objects in the Applies To list.

Hyperlink Required **Variant**. A Variant representing the data stored in a Hyperlink field.

Part Optional **AcHyperlinkPart** . The value for the *part* argument is an [intrinsic constant](#) representing the information you want returned by the **HyperlinkPart** method.

AcHyperlinkPart can be one of these AcHyperlinkPart constants.

acAddress. The *address* part of a Hyperlink field.

acDisplayedValue *default*. The underlined text displayed in a [hyperlink](#).

acDisplayText. The *displaytext* part of a Hyperlink field.

acFullAddress. The *address* and *subaddress* parts of a Hyperlink field delimited by a "#" character.

acScreenTip. The [tooltip](#) part of a Hyperlink field.

acSubAddress. The *subaddress* part of a Hyperlink field.

Remarks

You use the **HyperlinkPart** method to return one of three values from a Hyperlink field or the displayed value. The value returned depends on the setting of the *part* argument. The *part* argument is optional. If it's not used, the function returns the value Microsoft Access displays for the hyperlink (which corresponds to the **acDisplayedValue** setting for the *part* argument). The returned values can be one of the four parts of the Hyperlink field (*displaytext*, *address*, *subaddress*, or *screentip*), the full address, *address#subaddress*, or the value Microsoft Access displays for the hyperlink.

Note If you use the **HyperlinkPart** method in a query, the *part* argument is required and you can't use the constants listed above but must use the actual value instead.

When a value is provided in the *displaytext* part of a Hyperlink field, the value displayed by Microsoft Access will be the same as the *displaytext* setting. When there's no value in the *displaytext* part of a Hyperlink field, the value displayed will be the *address* or *subaddress* part of the Hyperlink field, depending on which value is first present in the field.

The following table shows the values returned by the **HyperlinkPart** method for data stored in a Hyperlink field.

Hyperlink field data	HyperlinkPart method returned values
	acDisplayedValue: http://www.microsoft.com
	acDisplayText:
	acAddress: http://www.microsoft.com
#http://www.microsoft.com#	acSubAddress:
	acScreenTip:

Microsoft#http://www.microsoft.com#

acFullAddress:
http://www.microsoft.com

acDisplayedValue: Microsoft

acDisplayText: Microsoft

acAddress:
http://www.microsoft.com

acSubAddress:

acScreenTip:

acFullAddress:
http://www.microsoft.com

acDisplayedValue: Customers

acDisplayText: Customers

acAddress:
http://www.microsoft.com

Customers#http://www.microsoft.com#Form
Customers

acSubAddress: Form Customers

acScreenTip:

acFullAddress:
http://www.microsoft.com#Form
Customer

acDisplayedValue: Form
Customers

acDisplayText:

##Form Customers#Enter Information

acAddress:

acSubAddress: Form Customers

acScreenTip: Enter Information

acFullAddress: #FormCustomer

When you add an *address* part to a Hyperlink field by using the **Insert Hyperlink** dialog box (available by clicking **Hyperlink** on the **Insert** menu) or by typing an address part directly into a Hyperlink field, Microsoft Access adds the two # symbols that delimit parts of the hyperlink data.

You can add or edit the *displaytext* part of a hyperlink field by right-clicking a hyperlink in a table, form, or report, pointing to **Hyperlink** on the shortcut menu, and then typing the display text in the **Text to display** box.

When you add data to a Hyperlink field directly, you must include the two # symbols to delimit the parts of the hyperlink data.

IMEStatus Function

Returns an Integer specifying the current Input Method Editor (IME) mode of Microsoft Windows; available in East Asian versions only.

Syntax

IMEStatus

Return Values

The return values for the Japanese locale are as follows:

Constant	Value	Description
vbIMEModeNoControl	0	Don't control IME (default)
vbIMEModeOn	1	IME on
vbIMEModeOff	2	IME off
vbIMEModeDisable	3	IME disabled
vbIMEModeHiragana	4	Full-width Hiragana mode
vbIMEModeKatakana	5	Full-width Katakana mode
vbIMEModeKatakanaHalf	6	Half-width Katakana mode
vbIMEModeAlphaFull	7	Full-width Alphanumeric mode
vbIMEModeAlpha	8	Half-width Alphanumeric mode

The return values for the Korean locale are as follows:

Constant	Value	Description
vbIMEModeNoControl	0	Don't control IME(default)
vbIMEModeAlphaFull	7	Full-width Alphanumeric mode
vbIMEModeAlpha	8	Half-width Alphanumeric mode
vbIMEModeHangulFull	9	Full-width Hangul mode
vbIMEModeHangul	10	Half-width Hangul mode

The return values for the Chinese locale are as follows:

Constant	Value	Description
vbIMEModeNoControl	0	Don't control IME (default)
vbIMEModeOn	1	IME on
vbIMEModeOff	2	IME off

Partition Function

Returns a **Variant (String)** indicating where a number occurs within a calculated series of ranges.

Syntax

Partition(*number, start, stop, interval*)

The **Partition** function syntax has these named arguments:

Part	Description
<i>number</i>	Required. Whole number that you want to evaluate against the ranges.
<i>start</i>	Required. Whole number that is the start of the overall range of numbers. The number can't be less than 0.
<i>stop</i>	Required. Whole number that is the end of the overall range of numbers. The number can't be equal to or less than <i>start</i> .

Remarks

The **Partition** function identifies the particular range in which *number* falls and returns a **Variant (String)** describing that range. The **Partition** function is most useful in queries. You can create a select query that shows how many orders fall within various ranges, for example, order values from 1 to 1000, 1001 to 2000, and so on.

The following table shows how the ranges are determined using three sets of *start*, *stop*, and *interval* parts. The First Range and Last Range columns show what **Partition** returns. The ranges are represented by *lowervalue:uppervalue*, where the low end (*lowervalue*) of the range is separated from the high end (*uppervalue*) of the range with a colon (:).

<i>start</i>	<i>stop</i>	<i>interval</i>	Before First	First Range	Last Range	After Last
0	99	5	" :-1"	" 0: 4"	" 95: 99"	" 100: "
20	199	10	" : 19"	" 20: 29"	" 190: 199"	" 200: "

100 1010 20 " : 99" " 100: 119" " 1000: 1010" " 1011: "

In the table shown above, the third line shows the result when *start* and *stop* define a set of numbers that can't be evenly divided by *interval*. The last range extends to *stop* (11 numbers) even though *interval* is 20.

If necessary, **Partition** returns a range with enough leading spaces so that there are the same number of characters to the left and right of the colon as there are characters in *stop*, plus one. This ensures that if you use **Partition** with other numbers, the resulting text will be handled properly during any subsequent sort operation.

If *interval* is 1, the range is *number:number*, regardless of the *start* and *stop* arguments. For example, if *interval* is 1, *number* is 100 and *stop* is 1000, **Partition** returns " 100: 100".

If any of the parts is Null, **Partition** returns a **Null**.

[Show All](#)

SysCmd Method

You can use the **SysCmd** method to, display a progress meter or optional specified text in the status bar, return information about Microsoft Access and its associated files, or return the state of a specified [database object](#) (to indicate whether the object is open, is a new object, or has been changed but not saved).

Variant.

expression.**SysCmd**(*Action*, *Argument2*, *Argument3*)

expression Required. An expression that returns one of the objects in the Applies To list.

Action Required **AcSysCmdAction** . One of the following [intrinsic constants](#) identifying the type of action to take. The following set of constants applies to a progress meter. The **SysCmd** method returns a **Null** if these actions are successful. Otherwise, Microsoft Access generates a [run-time error](#).

AcSysCmdAction can be one of these AcSysCmdAction constants.

acSysCmdAccessDir. Returns the name of the directory where Msaccess.exe is located.

acSysCmdAccessVer. Returns the version number of Microsoft Access.

acSysCmdClearHelpTopic

acSysCmdClearStatus. The following constant provides information on the state of a database object.

acSysCmdGetObjectState. Returns the state of the specified database object. You must specify *argument1* and *argument2* when you use this *action* value.

acSysCmdGetWorkgroupFile. Returns the path to the workgroup file (System.mdw).

acSysCmdIniFile. Returns the name of the .ini file associated with Microsoft Access.

acSysCmdInitMeter. Initializes the progress meter. You must specify the *argument1* and *argument2* arguments when you use this action.

acSysCmdProfile. Returns the **/profile** setting specified by the user when starting Microsoft Access from the command line.

acSysCmdRemoveMeter. Removes the progress meter.

acSysCmdRuntime. Returns **True** (-1) if a run-time version of Microsoft Access is running.

acSysCmdSetStatus. Sets the status bar text to the *text* argument.

acSysCmdUpdateMeter. Updates the progress meter with the specified value. You must specify the *text* argument when you use this action.

Argument2 Optional **Variant**. A [string expression](#) identifying the text to be displayed left-aligned in the status bar. This argument is required when the *action* argument is **acSysCmdInitMeter**, **acSysCmdUpdateMeter**, or **acSysCmdSetStatus**; this argument isn't valid for other *action* argument values.

Note When you specify the **acSysCmdGetObjectState** value for the *Action* parameter, then you must specify the appropriate [acObjectType](#).

acTable

acQuery

acForm

acReport

acMacro

acModule

acDataAccessPage

acDefault

acDiagram

acServerView

acStoreProcedure

This argument isn't valid for other *action* argument values.

Argument3 Optional **Variant**. A [numeric expression](#) that controls the display

of the progress meter. This argument is required when the *action* argument is **acSysCmdInitMeter**; this argument isn't valid for other *action* argument values.

Note When you specify the **acSysCmdGetObjectState** value for the *Action* parameter, then you must specify the name of the database object.

Remarks

For example, if you are building a custom wizard that creates a new form, you can use the **SysCmd** method to display a progress meter indicating the progress of your wizard as it constructs the form.

By calling the **SysCmd** method with the various progress meter actions, you can display a progress meter in the status bar for an operation that has a known duration or number of steps, and update it to indicate the progress of the operation.

To display a progress meter in the status bar, you must first call the **SysCmd** method with the **acSysCmdInitMeter** *action* argument, and the *text* and *value* arguments. When the *action* argument is **acSysCmdInitMeter**, the *value* argument is the maximum value of the meter, or 100 percent.

To update the meter to show the progress of the operation, call the **SysCmd** method with the **acSysCmdUpdateMeter** *action* argument and the *value* argument. When the *action* argument is **acSysCmdUpdateMeter**, the **SysCmd** method uses the *value* argument to calculate the percentage displayed by the meter. For example, if you set the maximum value to 200 and then update the meter with a value of 100, the progress meter will be half-filled.

You can also change the text that's displayed in the status bar by calling the **SysCmd** method with the **acSysCmdSetStatus** *action* argument and the *text* argument. For example, during a sort you might change the text to "Sorting...". When the sort is complete, you would reset the status bar by removing the text. The *text* argument can contain approximately 80 characters. Because the status bar text is displayed by using a proportional font, the actual number of characters you can display is determined by the total width of all the characters specified by the *text* argument.

As you increase the width of the status bar text, you decrease the length of the meter. If the text is longer than the status bar and the *action* argument is **acSysCmdInitMeter**, the **SysCmd** method ignores the text and doesn't display anything in the status bar. If the text is longer than the status bar and the *action* argument is **acSysCmdSetStatus**, the **SysCmd** method truncates the text to fit the status bar.

You can't set the status bar text to a [zero-length string](#) (" "). If you want to remove the existing text from the status bar, set the *text* argument to a single space. The following examples illustrate ways to remove the text from the status bar:

```
varReturn = SysCmd(acSysCmdInitMeter, " ", 100)
varReturn = SysCmd(acSysCmdSetStatus, " ")
```

If the progress meter is already displayed when you set the text by calling the **SysCmd** method with the **acSysCmdSetStatus** *action* argument, the **SysCmd** method automatically removes the meter.

Call the **SysCmd** method with other actions to determine system information about Microsoft Access, including which version number of Microsoft Access is running, whether it is a run-time version, the location of the Microsoft Access executable file, the setting for the **/profile** argument specified in the command line, and the name of an .ini file associated with Microsoft Access.

Note Both general and customized settings for Microsoft Access are now stored in the Windows Registry, so you probably won't need an .ini file with your Microsoft Access application. The **acSysCmdIniFile** *action* argument exists for compatibility with earlier versions of Microsoft Access.

Call the **SysCmd** method with the **acSysCmdGetObjectState** *action* argument and the *objecttype* and *objectname* arguments to return the state of a specified database object. An object can be in one of four possible states: not open or nonexistent, open, new, or changed but not saved.

For example, if you are designing a wizard that inserts a new field in a table, you need to determine whether the structure of the table has been changed but not yet saved, so that you can save it before modifying its structure. You can check the value returned by the **SysCmd** method to determine the state of the table.

The **SysCmd** method with the **acSysCmdGetObjectState** *action* argument can return any combination of the following constants.

Constant	State of database object	Value
acObjStateOpen	Open	1
acObjStateDirty	Changed but not saved	2

Note If the object referred to by the *objectname* argument is either not open or doesn't exist, the **SysCmd** method returns a value of zero.

The following code can be used to enable the use of your ActiveX control in expressions when the ActiveX control has been added to a form:

```
SysCmd 14, "<ActiveX Control GUID>"
```

Note Replace <ActiveX Control GUID> with the globally unique identifier (GUID) that identifies the ActiveX control that you want to enable in expressions.

Note You cannot remove an ActiveX control once it has been added to the list of allowed controls.

Date Function

Returns a **Variant (Date)** containing the current system date.

Syntax

Date

Remarks

To set the system date, use the **Date** statement.

Date, and if the calendar is Gregorian, **Date\$** behavior is unchanged by the **Calendar** property setting. If the calendar is Hijri, **Date\$** returns a 10-character string of the form *mm-dd-yyyy*, where *mm* (01-12), *dd* (01-30) and *yyyy* (1400-1523) are the Hijri month, day and year. The equivalent Gregorian range is Jan 1, 1980 through Dec 31, 2099.

DateAdd Function

Returns a **Variant (Date)** containing a date to which a specified time interval has been added.

Syntax

DateAdd(*interval*, *number*, *date*)

The **DateAdd** function syntax has these named arguments:

Part	Description
<i>interval</i>	Required. String expression that is the interval of time you want to add.
<i>number</i>	Required. Numeric expression that is the number of intervals you want to add. It can be positive (to get dates in the future) or negative (to get dates in the past).
<i>date</i>	Required. Variant (Date) or literal representing date to which the interval is added.

Settings

The *interval* argument has these settings:

Setting Description

yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week
h	Hour
n	Minute

s Second

Remarks

You can use the **DateAdd** function to add or subtract a specified time interval from a date. For example, you can use **DateAdd** to calculate a date 30 days from today or a time 45 minutes from now.

To add days to *date*, you can use Day of Year ("y"), Day ("d"), or Weekday ("w").

The **DateAdd** function won't return an invalid date. The following example adds one month to January 31:

```
DateAdd("m", 1, "31-Jan-95")
```

In this case, **DateAdd** returns 28-Feb-95, not 31-Feb-95. If *date* is 31-Jan-96, it returns 29-Feb-96 because 1996 is a leap year.

If the calculated date would precede the year 100 (that is, you subtract more years than are in *date*), an error occurs.

If *number* isn't a Long value, it is rounded to the nearest whole number before being evaluated.

Note The format of the return value for **DateAdd** is determined by **Control Panel** settings, not by the format that is passed in *date* argument.

Note For *date*, if the **Calendar** property setting is Gregorian, the supplied date must be Gregorian. If the calendar is Hijri, the supplied date must be Hijri. If month values are names, the name must be consistent with the current **Calendar** property setting. To minimize the possibility of month names conflicting with the current **Calendar** property setting, enter numeric month values (Short Date format).

DateDiff Function

Returns a **Variant (Long)** specifying the number of time intervals between two specified dates.

Syntax

DateDiff(*interval*, *date1*, *date2*[, *firstdayofweek*[, *firstweekofyear*]])

The **DateDiff** function syntax has these named arguments:

Part	Description
<i>interval</i>	Required. String expression that is the interval of time you use to calculate the difference between <i>date1</i> and <i>date2</i> .
<i>date1</i> , <i>date2</i>	Required; Variant (Date) . Two dates you want to use in the calculation.
<i>firstdayofweek</i>	Optional. A constant that specifies the first day of the week. If not specified, Sunday is assumed.
<i>firstweekofyear</i>	Optional. A constant that specifies the first week of the year. If not specified, the first week is assumed to be the week in which January 1 occurs.

Settings

The *interval* argument has these settings:

Setting Description

yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week

h	Hour
n	Minute
s	Second

The *firstdayofweek* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use the NLS API setting.
vbSunday	1	Sunday (default)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

Constant	Value	Description
vbUseSystem	0	Use the NLS API setting.
vbFirstJan1	1	Start with week in which January 1 occurs (default).
vbFirstFourDays	2	Start with the first week that has at least four days in the new year.
vbFirstFullWeek	3	Start with first full week of the year.

Remarks

You can use the **DateDiff** function to determine how many specified time intervals exist between two dates. For example, you might use **DateDiff** to calculate the number of days between two dates, or the number of weeks between today and the end of the year.

To calculate the number of days between *date1* and *date2*, you can use either Day of year ("y") or Day ("d"). When *interval* is Weekday ("w"), **DateDiff** returns the number of weeks between the two dates. If *date1* falls on a Monday, **DateDiff** counts the number of Mondays until *date2*. It counts *date2* but not

date1. If *interval* is Week ("ww"), however, the **DateDiff** function returns the number of calendar weeks between the two dates. It counts the number of Sundays between *date1* and *date2*. **DateDiff** counts *date2* if it falls on a Sunday; but it doesn't count *date1*, even if it does fall on a Sunday.

If *date1* refers to a later point in time than *date2*, the **DateDiff** function returns a negative number.

The *firstdayofweek* argument affects calculations that use the "w" and "ww" interval symbols.

If *date1* or *date2* is a date literal, the specified year becomes a permanent part of that date. However, if *date1* or *date2* is enclosed in double quotation marks (" "), and you omit the year, the current year is inserted in your code each time the *date1* or *date2* expression is evaluated. This makes it possible to write code that can be used in different years.

When comparing December 31 to January 1 of the immediately succeeding year, **DateDiff** for Year ("yyyy") returns 1 even though only a day has elapsed.

Note For *date1* and *date2*, if the **Calendar** property setting is Gregorian, the supplied date must be Gregorian. If the calendar is Hijri, the supplied date must be Hijri.

DatePart Function

Returns a **Variant (Integer)** containing the specified part of a given date.

Syntax

DatePart(*interval*, *date*[, *firstdayofweek*[, *firstweekofyear*]])

The **DatePart** function syntax has these named arguments:

Part	Description
<i>interval</i>	Required. String expression that is the interval of time you want to return.
<i>date</i>	Required. Variant (Date) value that you want to evaluate.
<i>firstdayofweek</i>	Optional. A constant that specifies the first day of the week. If not specified, Sunday is assumed.
<i>firstweekofyear</i>	Optional. A constant that specifies the first week of the year. If not specified, the first week is assumed to be the week in which January 1 occurs.

Settings

The *interval* argument has these settings:

Setting Description

yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week
h	Hour

n	Minute
s	Second

The *firstdayofweek* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use the NLS API setting.
vbSunday	1	Sunday (default)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

The *firstweekofyear* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use the NLS API setting.
vbFirstJan1	1	Start with week in which January 1 occurs (default).
vbFirstFourDays	2	Start with the first week that has at least four days in the new year.
vbFirstFullWeek	3	Start with first full week of the year.

Remarks

You can use the **DatePart** function to evaluate a date and return a specific interval of time. For example, you might use **DatePart** to calculate the day of the week or the current hour.

The *firstdayofweek* argument affects calculations that use the "w" and "ww" interval symbols.

If *date* is a date literal, the specified year becomes a permanent part of that date.

However, if *date* is enclosed in double quotation marks (" "), and you omit the year, the current year is inserted in your code each time the *date* expression is evaluated. This makes it possible to write code that can be used in different years.

Note For *date*, if the **Calendar** property setting is Gregorian, the supplied date must be Gregorian. If the calendar is Hijri, the supplied date must be Hijri.

The returned date part is in the time period units of the current Arabic calendar. For example, if the current calendar is Hijri and the date part to be returned is the year, the year value is a Hijri year.

IsDate Function

Returns a **Boolean** value indicating whether an expression can be converted to a date.

Syntax

IsDate(*expression*)

The required *expression* argument is a Variant containing a date expression or string expression recognizable as a date or time.

Remarks

IsDate returns **True** if the expression is a date or is recognizable as a valid date; otherwise, it returns **False**. In Microsoft Windows, the range of valid dates is January 1, 100 A.D. through December 31, 9999 A.D.; the ranges vary among operating systems.

MonthName Function

Description

Returns a string indicating the specified month.

Syntax

MonthName(*month*[, *abbreviate*])

The **MonthName** function syntax has these parts:

Part	Description
<i>month</i>	Required. The numeric designation of the month. For example, January is 1, February is 2, and so on.
<i>abbreviate</i>	Optional. Boolean value that indicates if the month name is to be abbreviated. If omitted, the default is False , which means that the month name is not abbreviated.

Now Function

Returns a **Variant (Date)** specifying the current date and time according your computer's system date and time.

Syntax

Now

Time Function

Returns a **Variant (Date)** indicating the current system time.

Syntax

Time

Remarks

To set the system time, use the **Time** statement.

Timer Function

Returns a **Single** representing the number of seconds elapsed since midnight.

Syntax

Timer

Remarks

In Microsoft Windows the **Timer** function returns fractional portions of a second. On the Macintosh, timer resolution is one second.

WeekdayName Function

Description

Returns a string indicating the specified day of the week.

Syntax

WeekdayName(*weekday*, *abbreviate*, *firstdayofweek*)

The **WeekdayName** function syntax has these parts:

Part	Description
<i>weekday</i>	Required. The numeric designation for the day of the week. Numeric value of each day depends on setting of the <i>firstdayofweek</i> setting.
<i>abbreviate</i>	Optional. Boolean value that indicates if the weekday name is to be abbreviated. If omitted, the default is False , which means that the weekday name is not abbreviated.
<i>firstdayofweek</i>	Optional. Numeric value indicating the first day of the week. See Settings section for values.

Settings

The *firstdayofweek* argument can have the following values:

Constant	Value	Description
vbUseSystem	0	Use National Language Support (NLS) API setting.
vbSunday	1	Sunday (default)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

[Show All](#)

DDE Function

You can use the **DDE** function to initiate a [dynamic data exchange](#) (DDE) conversation with another application, request an item of information from that application, and display that information in a control on a form or report.

For example, you can use the **DDE** function in the **ControlSource** property of a text box to display data from a specified cell in a Microsoft Excel spreadsheet.

DDE(application, topic, item)

The **DDE** function has the following arguments.

Argument	Description
<i>application</i>	A string expression identifying an application that can participate in a DDE conversation. Usually, <i>application</i> is the name of an .exe file (without the .exe extension) for a Microsoft Windows–based application, such as Microsoft Excel. For example, to initiate a DDE conversation with Microsoft Excel, type " Excel " for the <i>application</i> argument.
<i>topic</i>	A string expression that is the name of a topic recognized by <i>application</i> . The <i>topic</i> argument is often a document or data file. Check the other application's documentation for a list of possible topics.
<i>item</i>	A string expression that is the name of a data item recognized by <i>application</i> . Check the other application's documentation for a list of possible items.

Remarks

The **DDE** function attempts to initiate a DDE conversation with the specified *application* and *topic*, and requests the data in *item*. If successful, the **DDE** function returns a string that contains the requested information.

If you are requesting data from Microsoft Excel, *item* might be a row-and-column identifier, such as "R1C1" or the name of a range of cells. In the following example, the **DDE** function requests information from the cell at row 1, column 1 in a Microsoft Excel worksheet. You can enter this expression for a text box control in the **ControlSource** property box on the control's property sheet:

```
=DDE("Excel", "Sheet1", "R1C1")
```

You can use the **DDE** function only in the **ControlSource** property of a text box, option group, check box, or combo box. You can't call the **DDE** function from Visual Basic.

When you use the **DDE** function, the control becomes read-only in [Form view](#) and [Print Preview](#). For example, if you use the **DDE** function in a text box, the text in the text box can't be edited. You must edit the text in the other application. Because the **ControlSource** property is read-only in Form view and Print Preview, changes to the control must be made in [Design view](#).

Microsoft Windows and your computer's memory and resources determine the maximum number of DDE conversations that can be open simultaneously. If the conversation can't be initiated because the other application isn't running or doesn't recognize the specified topic, or if the maximum number of conversations has already been reached, the **DDE** function returns a **Null**.

Note The other application may be configured to ignore your request for a DDE conversation. If so, the **DDE** function returns a **Null**. Similarly, you can set Microsoft Access to ignore requests from other applications: click **Options** on the **Tools** menu, and on the **Advanced** tab of the **Options** dialog box under **DDE operations**, select **Ignore DDE requests**.



If you need to manipulate another application's objects from Microsoft Access, you may want to consider using Automation.

The following table illustrates how the **DDE** function behaves when you use it with each of the [controls](#).

Control	Remarks
Text box	The <i>item</i> argument can refer to text or numbers. If <i>item</i> refers to more than one piece of information, such as a named range in a Microsoft Excel worksheet that contains multiple cells, the DDE function returns the first entry. You could use this function with a text box to display the data contained in a cell on the worksheet.
Combo box	The DDE function fills the combo box with the information referred to by <i>item</i> . You can't enter data in the text portion of the box. You might use the DDE function with a combo box to display a list of regions or countries that you maintain in a Microsoft Excel worksheet.
Option group	The OptionValue property of each option button in an option group is set to a number. Usually, the first button value is 1, the second is 2, and so on. The number returned by the DDE function determines which option button will be selected. For example, if the DDE function returns 2, the second button will be selected. If this function returns a value that doesn't match any of the OptionValue property settings, none of the buttons will be selected. If <i>item</i> refers to more than one piece of information, such as a named range in a Microsoft Excel worksheet that contains multiple cells, the DDE function returns the first entry.
Check box	If the DDE function returns 0, the check box will be cleared. If this function returns a nonzero number, such as 1 or -1, the box will be selected. If <i>item</i> refers to text or to more than one piece of information, such as a named range in a Microsoft Excel worksheet that contains multiple cells, the check box will be unavailable.

[Show All](#)

DDEInitiate Function

You can use the **DDEInitiate** function to begin a [dynamic data exchange \(DDE\)](#) conversation with another application. The **DDEInitiate** function opens a [DDE channel](#) for transfer of data between a DDE server and client application.

For example, if you wish to transfer data from a Microsoft Excel spreadsheet to a Microsoft Access database, you can use the **DDEInitiate** function to open a channel between the two applications. In this example, Microsoft Access acts as the client application and Microsoft Excel acts as the server application.

DDEInitiate(*application, topic*)

The **DDEInitiate** function has the following arguments.

Argument	Description
<i>application</i>	A string expression identifying an application that can participate in a DDE conversation. Usually, the <i>application</i> argument is the name of an .exe file (without the .exe extension) for a Microsoft Windows–based application, such as Microsoft Excel.
<i>topic</i>	A string expression that is the name of a topic recognized by the <i>application</i> argument. Check the application's documentation for a list of topics.

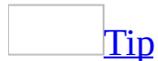
Remarks

If successful, the **DDEInitiate** function begins a DDE conversation with the application and topic specified by the *application* and *topic* arguments, and then returns a **Long** integer value. This return value represents a unique [channel number](#) identifying a channel through which data transfer can take place. This channel number is subsequently used with other DDE functions and statements.

If the application isn't already running or if it's running but doesn't recognize the *topic* argument or doesn't support DDE, the **DDEInitiate** function returns a [run-time error](#).

The value of the *topic* argument depends on the application specified by the *application* argument. For applications that use documents or data files, valid topic names often include the names of those files.

Note The maximum number of channels that can be open simultaneously is determined by Microsoft Windows and your computer's memory and resources. If you aren't using a channel, you should conserve resources by terminating it with a **DDETerminate** or **DDETerminateAll** statement.



If you need to manipulate another application's objects from Microsoft Access, you may want to consider using Automation.

[Show All](#)

DDERequest Function

You can use the **DDERequest** function over an open [dynamic data exchange \(DDE\)](#) channel to request an item of information from a DDE server application.

For example, if you have an open [DDE channel](#) between Microsoft Access and Microsoft Excel, you can use the **DDERequest** function to transfer text from a Microsoft Excel spreadsheet to a Microsoft Access database.

DDERequest(*channum*, *item*)

The **DDERequest** function has the following arguments.

Argument	Description
<i>channum</i>	A channel number , the integer returned by the DDEInitiate function.
<i>item</i>	A string expression that's the name of a data item recognized by the application specified by the DDEInitiate function. Check the application's documentation for a list of possible items.

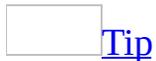
Remarks

The *channum* argument specifies the channel number of the desired DDE conversation, and the *item* argument identifies which data should be retrieved from the server application. The value of the *item* argument depends on the application and [topic](#) specified when the channel indicated by the *channum* argument is opened. For example, the *item* argument may be a range of cells in a Microsoft Excel spreadsheet.

The **DDERequest** function returns a [Variant](#) as a [string](#) containing the requested information if the request was successful.

The data is requested in alphanumeric text format. Graphics or text in any other format can't be transferred.

If the *channum* argument isn't an integer corresponding to an open channel, or if the data requested can't be transferred, a [run-time error](#) occurs.



If you need to manipulate another application's objects from Microsoft Access, you may want to consider using Automation.

[Show All](#)

LoadPicture Method

The **LoadPicture** method loads a graphic into an [ActiveX control](#).

expression.**LoadPicture**(*FileName*)

expression Required. An expression that returns one of the objects in the Applies To list.

FileName Required **String**. The file name of the graphic to be loaded. The graphic can be a bitmap file (.bmp), icon file (.ico), run-length encoded file (.rle), or metafile (.wmf).

Remarks

Assign the return value of the **LoadPicture** method to the **Picture** property of an ActiveX control to dynamically load a graphic into the control. The following example loads a bitmap into a control called OLECustomControl on an Orders form:

```
Set Forms!Orders!OLECustomControl.Picture = _  
    LoadPicture("Stars.bmp")
```

The **LoadPicture** method returns an object of type **Picture**. You can assign this value to a variable of type [Object](#) by using the **Set** statement.

The **Picture** object is not a [Microsoft Access object](#), but it is available to procedures in Microsoft Access.

Note You can't use the **LoadPicture** method to set the **Picture** property of an [image control](#). This method works with ActiveX controls only. To set the **Picture** property of an image control, simply assign to it a [string](#) specifying the file name and path of the desired graphic.

[Show All](#)

DAvg Function

You can use the **DAvg** function to calculate the average of a set of values in a specified set of records (a [domain](#)). Use the **DAvg** function in Visual Basic code or in a [macro](#), in a query expression, or in a [calculated control](#).

For example, you could use the **DAvg** function in the criteria row of a select query on freight cost to restrict the results to those records where the freight cost exceeds the average. Or you could use an expression including the **DAvg** function in a calculated control and display the average value of previous orders next to the value of a new order.

DAvg(*expr*, *domain*, [*criteria*])

The **DAvg** function has the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the field containing the numeric data you want to average. It can be a string expression identifying a field in a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DAvg function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DAvg function evaluates <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ; otherwise the DAvg function returns a Null .

Remarks

Records containing **Null** values aren't included in the calculation of the average.

Whether you use the **DAvg** function in a macro or module, in a query expression, or in a calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DAvg** function to specify criteria in the Criteria row of a query. For example, suppose you want to view a list of all products ordered in quantities above the average order quantity. You could create a query on the Orders, Order Details, and Products tables, and include the Product Name field and the Quantity field, with the following expression in the Criteria row beneath the Quantity field:

```
>DAvg("[Quantity]", "Orders")
```

You can also use the **DAvg** function within a calculated field expression in a query, or in the Update To row of an [update query](#).

Note You can use either the **DAvg** or **Avg** function in a calculated field expression in a [totals query](#). If you use the **DAvg** function, values are averaged before the data is grouped. If you use the **Avg** function, the data is grouped before values in the field expression are averaged.

Use the **DAvg** function in a calculated control when you need to specify criteria to restrict the range of data on which the **DAvg** function is performed. For example, to display the average cost of freight for shipments sent to California, set the **ControlSource** property of a text box to the following expression:

```
=DAvg("[Freight]", "Orders", "[ShipRegion] = 'CA'")
```

If you simply want to average all records in *domain*, use the **Avg** function.

You can use the **DAvg** function in a module or macro or in a calculated control on a form if a field that you need to display isn't in the record source on which your form is based. For example, suppose you have a form based on the Orders

table, and you want to include the Quantity field from the Order Details table in order to display the average number of items ordered by a particular customer. You can use the **DAvg** function to perform this calculation and display the data on your form.

Tips

- If you use the **DAvg** function in a calculated control, you may want to place the control on the form header or footer so that the value for this control is not recalculated each time you move to a new record.
- If the data type of the field from which *expr* is derived is a number, the **DAvg** function returns a **Double** data type. If you use the **DAvg** function in a calculated control, include a data type conversion function in the expression to improve performance.
- Although you can use the **DAvg** function to determine the average of values in a field in a [foreign table](#), it may be more efficient to create a query that contains all of the fields that you need and then base your form or report on that query.

Note Unsaved changes to records in *domain* aren't included when you use this function. If you want the **DAvg** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

[Show All](#)

DCount Function

You can use the **DCount** function to determine the number of records that are in a specified set of records (a [domain](#)). Use the **DCount** function in Visual Basic, a [macro](#), a query expression, or a [calculated control](#).

For example, you could use the **DCount** function in a module to return the number of records in an Orders table that correspond to orders placed on a particular date.

DCount(*expr*, *domain*, [*criteria*])

The **DCount** function has the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the field for which you want to count records. It can be a string expression identifying a field in a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DCount function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DCount function evaluates <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ; otherwise the DCount function returns a Null .

Remarks

Use the **DCount** function to count the number of records in a domain when you don't need to know their particular values. Although the *expr* argument can perform a calculation on a field, the **DCount** function simply tallies the number of records. The value of any calculation performed by *expr* is unavailable.

Use the **DCount** function in a calculated control when you need to specify criteria to restrict the range of data on which the function is performed. For example, to display the number of orders to be shipped to California, set the **ControlSource** property of a text box to the following expression:

```
=DCount("[OrderID]", "Orders", "[ShipRegion] = 'CA'")
```

If you simply want to count all records in *domain* without specifying any restrictions, use the **Count** function.



Tip

The **Count** function has been optimized to speed counting of records in queries. Use the **Count** function in a query expression instead of the **DCount** function, and set optional criteria to enforce any restrictions on the results. Use the **DCount** function when you must count records in a domain from within a code module or macro, or in a calculated control.

You can use the **DCount** function to count the number of records containing a particular field that isn't in the record source on which your form or report is based. For example, you could display the number of orders in the Orders table in a calculated control on a form based on the Products table.

The **DCount** function doesn't count records that contain **Null** values in the field referenced by *expr* unless *expr* is the asterisk (*) wildcard character. If you use an asterisk, the **DCount** function calculates the total number of records, including those that contain **Null** fields. The following example calculates the number of records in an Orders table.

```
intX = DCount("*", "Orders")
```

If *domain* is a table with a [primary key](#), you can also count the total number of records by setting *expr* to the primary key field, since there will never be a **Null** in the primary key field.

If *expr* identifies multiple fields, separate the field names with a concatenation operator, either an ampersand (&) or the addition operator (+). If you use an ampersand to separate the fields, the **DCount** function returns the number of records containing data in any of the listed fields. If you use the addition operator, the **DCount** function returns only the number of records containing data in all of the listed fields. The following example demonstrates the effects of each operator when used with a field that contains data in all records (ShipName) and a field that contains no data (ShipRegion).

```
intW = DCount("[ShipName]", "Orders")
intX = DCount("[ShipRegion]", "Orders")
intY = DCount("[ShipName] + [ShipRegion]", "Orders")
intZ = DCount("[ShipName] & [ShipRegion]", "Orders")
```

Note The ampersand is the preferred operator for performing string concatenation. You should avoid using the addition operator for anything other than numeric addition, unless you specifically wish to propagate **Nulls** through an expression.

Unsaved changes to records in *domain* aren't included when you use this function. If you want the **DCount** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

[Show All](#)

DFirst, DLast Functions

You can use the **DFirst** and **DLast** functions to return a random record from a particular field in a table or query when you simply need any value from that field. Use the **DFirst** and **DLast** functions in a [macro](#), module, query expression, or [calculated control](#) on a form or report.

DFirst(*expr*, *domain*, [*criteria*])

DLast(*expr*, *domain*, [*criteria*])

The **DFirst** and **DLast** functions have the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the field from which you want to find the first or last value. It can be either a string expression identifying a field in a table or query, or an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DFirst or DLast function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DFirst and DLast functions evaluate <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ; otherwise, the DFirst and DLast functions return a Null .

Remarks

Note If you want to return the first or last record in a set of records (a [domain](#)), you should create a query sorted as either ascending or descending and set the **TopValues** property to 1. For more information, see the **TopValues** property topic. From Visual Basic, you can also create an ADO **Recordset** object and use the **MoveFirst** or **MoveLast** method to return the first or last record in a set of records.

[Show All](#)

DLookup Function

You can use the **DLookup** function to get the value of a particular field from a specified set of records (a [domain](#)). Use the **DLookup** function in Visual Basic, a [macro](#), a query expression, or a [calculated control](#) on a form or report.

You can use the **DLookup** function to display the value of a field that isn't in the record source for your form or report. For example, suppose you have a form based on an Order Details table. The form displays the OrderID, ProductID, UnitPrice, Quantity, and Discount fields. However, the ProductName field is in another table, the Products table. You could use the **DLookup** function in a calculated control to display the ProductName on the same form.

DLookup(*expr*, *domain*, [*criteria*])

The **DLookup** function has the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the field whose value you want to return. It can be a string expression identifying a field in a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DLookup function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DLookup function evaluates <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ; otherwise, the DLookup function returns a Null .

Remarks

The **DLookup** function returns a single field value based on the information specified in *criteria*. Although *criteria* is an optional argument, if you don't supply a value for *criteria*, the **DLookup** function returns a random value in the domain.

If no record satisfies *criteria* or if *domain* contains no records, the **DLookup** function returns a **Null**.

If more than one field meets *criteria*, the **DLookup** function returns the first occurrence. You should specify criteria that will ensure that the field value returned by the **DLookup** function is unique. You may want to use a [primary key](#) value for your criteria, such as [EmployeeID] in the following example, to ensure that the **DLookup** function returns a unique value:

```
Dim varX As Variant
varX = DLookup("[LastName]", "Employees", "[EmployeeID] = 1")
```

Whether you use the **DLookup** function in a macro or module, a query expression, or a calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DLookup** function to specify criteria in the Criteria row of a query, within a calculated field expression in a query, or in the Update To row in an [update query](#).

You can also use the **DLookup** function in an expression in a calculated control on a form or report if the field that you need to display isn't in the record source on which your form or report is based. For example, suppose you have an Order Details form based on an Order Details table with a text box called ProductID that displays the ProductID field. To look up ProductName from a Products table based on the value in the text box, you could create another text box and set its **ControlSource** property to the following expression:

```
=DLookup("[ProductName]", "Products", "[ProductID] =" & _
    & Forms![Order Details]!ProductID)
```

Tips

- Although you can use the **DLookup** function to display a value from a field in a [foreign table](#), it may be more efficient to create a query that contains the fields that you need from both tables and then to base your form or report on that query.
- You can also use the Lookup Wizard to find values in a foreign table.

Note Unsaved changes to records in *domain* aren't included when you use this function. If you want the **DLookup** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

Example

The following example returns name information from the `CompanyName` field of the record satisfying *criteria*. The domain is a `Shippers` table. The *criteria* argument restricts the resulting set of records to those for which `ShipperID` equals 1.

```
Dim varX As Variant
varX = DLookup("[CompanyName]", "Shippers", "[ShipperID] = 1")
```

The next example from the `Shippers` table uses the form control `ShipperID` to provide criteria for the **DLookup** function. Note that the reference to the control isn't included in the quotation marks that denote the strings. This ensures that each time the **DLookup** function is called, Microsoft Access will obtain the current value from the control.

```
Dim varX As Variant
varX = DLookup("[CompanyName]", "Shippers", "[ShipperID] = " _
    & Forms!Shippers!ShipperID)
```

The next example uses a variable, `intSearch`, to get the value.

```
Dim intSearch As Integer
Dim varX As Variant

intSearch = 1
varX = DLookup("[CompanyName]", "Shippers", _
    "[ShipperID] = " & intSearch)
```

[Show All](#)

DMin, DMax Functions

You can use the **DMin** and **DMax** functions to determine the minimum and maximum values in a specified set of records (a [domain](#)). Use the **DMin** and **DMax** functions in Visual Basic, a [macro](#), a query expression, or a [calculated control](#).

For example, you could use the **DMin** and **DMax** functions in calculated controls on a report to display the smallest and largest order amounts for a particular customer. Or you could use the **DMin** function in a query expression to display all orders with a discount greater than the minimum possible discount.

DMin(*expr*, *domain*, [*criteria*])

DMax(*expr*, *domain*, [*criteria*])

The **DMin** and **DMax** functions have the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the field for which you want to find the minimum or maximum value. It can be a string expression identifying a field in a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DMin or DMax function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DMin and DMax functions evaluate <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ,

otherwise the **DMin** and **DMax** functions returns a **[Null](#)**.

Remarks

The **DMin** and **DMax** functions return the minimum and maximum values that satisfy *criteria*. If *expr* identifies numeric data, the **DMin** and **DMax** functions return numeric values. If *expr* identifies string data, they return the string that is first or last alphabetically.

The **DMin** and **DMax** functions ignore **Null** values in the field referenced by *expr*. However, if no record satisfies *criteria* or if *domain* contains no records, the **DMin** and **DMax** functions return a **Null**.

Whether you use the **DMin** or **DMax** function in a macro, module, query expression, or calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DMin** and **DMax** function to specify criteria in the Criteria row of a query, in a calculated field expression in a query, or in the Update To row of an [update query](#).

Note You can use the **DMin** and **DMax** functions or the **Min** and **Max** functions in a calculated field expression of a [totals query](#). If you use the **DMin** or **DMax** function, values are evaluated before the data is grouped. If you use the **Min** or **Max** function, the data is grouped before values in the field expression are evaluated.

Use the **DMin** or **DMax** function in a calculated control when you need to specify criteria to restrict the range of data on which the function is performed. For example, to display the maximum freight charged for an order shipped to California, set the **ControlSource** property of a text box to the following expression:

```
=DMax("[Freight]", "Orders", "[ShipRegion] = 'CA'")
```

If you simply want to find the minimum or maximum value of all records in *domain*, use the **Min** or **Max** function.

You can use the **DMin** or **DMax** function in a module or macro or in a calculated control on a form if the field that you need to display is not in the

record source on which your form is based.



Although you can use the **DMin** or **DMax** function to find the minimum or maximum value from a field in a [foreign table](#), it may be more efficient to create a query that contains the fields that you need from both tables and base your form or report on that query.

Note Unsaved changes to records in *domain* aren't included when you use these functions. If you want the **DMax** or **DMin** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.



[Show All](#)

DStDev, DStDevP Functions

You can use the **DStDev** and **DStDevP** functions to estimate the standard deviation across a set of values in a specified set of records (a [domain](#)). Use the **DStDev** and **DStDevP** functions in Visual Basic, a [macro](#), a query expression, or a [calculated control](#) on a form or report.

Use the **DStDevP** function to evaluate a population and the **DStDev** function to evaluate a population sample.

For example, you could use the **DStDev** function in a module to calculate the standard deviation across a set of students' test scores.

DStDev(*expr*, *domain*, [*criteria*])

DStDevP(*expr*, *domain*, [*criteria*])

The **DStDev** and **DStDevP** functions have the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the numeric field on which you want to find the standard deviation. It can be a string expression identifying a field from a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DStDev or DStDevP function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DStDev and DStDevP functions evaluate <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in

domain; otherwise, the **DStDev** and **DStDevP** functions will return a **Null**.

Remarks

If *domain* refers to fewer than two records or if fewer than two records satisfy *criteria*, the **DStDev** and **DStDevP** functions return a **Null**, indicating that a standard deviation can't be calculated.

Whether you use the **DStDev** or **DStDevP** function in a macro, module, query expression, or calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DStDev** and **DStDevP** functions to specify criteria in the Criteria row of a select query. For example, you could create a query on an Orders table and a Products table to display all products for which the freight cost fell above the mean plus the standard deviation for freight cost. The Criteria row beneath the Freight field would contain the following expression:

```
>(DStDev("[Freight]", "Orders") + DAvg("[Freight]", "Orders"))
```

You can use the **DStDev** and **DStDevP** functions in a calculated field expression of a query, or in the Update To row of an [update query](#).

Note You can use the **DStDev** and **DStDevP** functions or the **StDev** and **StDevP** functions in a calculated field expression of a [totals query](#). If you use the **DStDev** or **DStDevP** function, values are calculated before data is grouped. If you use the **StDev** or **StDevP** function, the data is grouped before values in the field expression are evaluated.

Use the **DStDev** and **DStDevP** function in a calculated control when you need to specify criteria to restrict the range of data on which the function is performed. For example, to display standard deviation for orders to be shipped to California, set the **ControlSource** property of a text box to the following expression:

```
=DStDev("[Freight]", "Orders", "[ShipRegion] = 'CA'")
```

If you simply want to find the standard deviation across all records in *domain*, use the **StDev** or **StDevP** function.



Tip

If the data type of the field from which *expr* is derived is a number, the **DStDev** and **DStDevP** functions return a **Double** data type. If you use the **DStDev** or **DStDevP** function in a calculated control, include a data type conversion function in the expression to improve performance.

Note Unsaved changes to records in *domain* are not included when you use these functions. If you want the **DStDev** or **DStDevP** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

[Show All](#)

DSum Function

You can use the **DSum** functions to calculate the sum of a set of values in a specified set of records (a [domain](#)). Use the **DSum** function in Visual Basic, a [macro](#), a query expression, or a [calculated control](#).

For example, you could use the **DSum** function in a calculated field expression in a query to calculate the total sales made by a particular employee over a period of time. Or you could use the **DSum** function in a calculated control to display a running sum of sales for a particular product.

DSum(*expr*, *domain*, [*criteria*])

The **DSum** function has the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the numeric field whose values you want to total. It can be a string expression identifying a field in a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name of a field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DSum function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DSum function evaluates <i>expr</i> against the entire domain. Any field that is included in <i>criteria</i> must also be a field in <i>domain</i> ; otherwise, the DSum function returns a Null .

Remarks

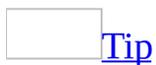
If no record satisfies the *criteria* argument or if domain contains no records, the **DSum** function returns a **Null**.

Whether you use the **DSum** function in a macro, module, query expression, or calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DSum** function to specify criteria in the Criteria row of a query, in a calculated field in a query expression, or in the Update To row of an [update query](#).

Note You can use either the **DSum** or **Sum** function in a calculated field expression in a [totals query](#). If you use the **DSum** function, values are calculated before data is grouped. If you use the **Sum** function, the data is grouped before values in the field expression are evaluated.

You may want to use the **DSum** function when you need to display the sum of a set of values from a field that is not in the record source for your form or report. For example, suppose you have a form that displays information about a particular product. You could use the **DSum** function to maintain a running total of sales of that product in a calculated control.



If you need to maintain a running total in a control on a report, you can use the **RunningSum** property of that control if the field on which it is based is included in the record source for the report. Use the **DSum** function to maintain a running sum on a form.

Note Unsaved changes to records in *domain* aren't included when you use this function. If you want the **DSum** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

[Show All](#)

DVar, DVarP functions

You can use the **DVar** and **DVarP** functions to estimate variance across a set of values in a specified set of records (a [domain](#)). Use the **DVar** and **DVarP** functions in Visual Basic, a [macro](#), a query expression, or a [calculated control](#) on a form or report.

Use the **DVarP** function to evaluate variance across a population and the **DVar** function to evaluate variance across a population sample.

For example, you could use the **DVar** function to calculate the variance across a set of students' test scores.

DVar(*expr*, *domain*, [*criteria*])

DVarP(*expr*, *domain*, [*criteria*])

The **DVar** and **DVarP** functions have the following arguments.

Argument	Description
<i>expr</i>	An expression that identifies the numeric field on which you want to find the variance. It can be a string expression identifying a field from a table or query, or it can be an expression that performs a calculation on data in that field. In <i>expr</i> , you can include the name field in a table, a control on a form, a constant, or a function. If <i>expr</i> includes a function, it can be either built-in or user-defined, but not another domain aggregate or SQL aggregate function. Any field included in <i>expr</i> must be a numeric field.
<i>domain</i>	A string expression identifying the set of records that constitutes the domain. It can be a table name or a query name for a query that does not require a parameter.
<i>criteria</i>	An optional string expression used to restrict the range of data on which the DVar or DVarP function is performed. For example, <i>criteria</i> is often equivalent to the WHERE clause in an SQL expression, without the word WHERE. If <i>criteria</i> is omitted, the DVar and DVarP functions evaluate <i>expr</i> against the entire domain.

Any field that is included in *criteria* must also be a field in *domain*; otherwise the **DVar** and **DVarP** functions return a [Null](#).

Remarks

If *domain* refers to fewer than two records or if fewer than two records satisfy *criteria*, the **DVar** and **DVarP** functions return a **Null**, indicating that a variance can't be calculated.

Whether you use the **DVar** or **DVarP** function in a macro, module, query expression, or calculated control, you must construct the *criteria* argument carefully to ensure that it will be evaluated correctly.

You can use the **DVar** and **DVarP** function to specify criteria in the **Criteria** row of a select query, in a calculated field expression in a query, or in the **Update To** row of an update query.

Note You can use the **DVar** and **DVarP** functions or the **Var** and **VarP** functions in a calculated field expression in a [totals query](#). If you use the **DVar** or **DVarP** function, values are calculated before data is grouped. If you use the **Var** or **VarP** function, the data is grouped before values in the field expression are evaluated.

Use the **DVar** and **DVarP** functions in a calculated control when you need to specify *criteria* to restrict the range of data on which the function is performed. For example, to display a variance for orders to be shipped to California, set the **ControlSource** property of a text box to the following expression:

```
=DVar("[Freight]", "Orders", "[ShipRegion] = 'CA'")
```

If you simply want to find the standard deviation across all records in *domain*, use the **Var** or **VarP** function.

Note Unsaved changes to records in *domain* are not included when you use these functions. If you want the **DVar** or **DVarP** function to be based on the changed values, you must first save the changes by clicking **Save Record** on the **Records** menu, moving the focus to another record, or by using the **Update** method.

CVErr Function

Returns a Variant of subtype **Error** containing an error number specified by the user.

Syntax

CVErr(*errornumber*)

The required *errornumber* argument is any valid error number.

Remarks

Use the **CVErr** function to create user-defined errors in user-created procedures. For example, if you create a function that accepts several arguments and normally returns a string, you can have your function evaluate the input arguments to ensure they are within acceptable range. If they are not, it is likely your function will not return what you expect. In this event, **CVErr** allows you to return an error number that tells you what action to take.

Note that implicit conversion of an **Error** is not allowed. For example, you can't directly assign the return value of **CVErr** to a variable that is not a **Variant**. However, you can perform an explicit conversion (using **CInt**, **Cdbl**, and so on) of the value returned by **CVErr** and assign that to a variable of the appropriate data type.

Err Object

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Contains information about run-time errors.

Remarks

The properties of the **Err** object are set by the generator of an error— Visual Basic, an object, or the programmer.

The default property of the **Err** object is **Number**. Because the default property can be represented by the object name **Err**, earlier code written using the **Err** function or **Err** statement doesn't have to be modified.

When a run-time error occurs, the properties of the **Err** object are filled with information that uniquely identifies the error and information that can be used to handle it. To generate a run-time error in your code, use the **Raise** method.

The **Err** object's properties are reset to zero or zero-length strings (""), after an **Exit Sub**, **Exit Function**, **Exit Property** or **Resume Next** statement within an error-handling routine. Using any form of the **Resume** statement outside of an error-handling routine will not reset the **Err** object's properties. The **Clear** method can be used to explicitly reset **Err**.

Use the **Raise** method, rather than the **Error** statement, to generate run-time errors for system errors and class modules. Using the **Raise** method in other code depends on the richness of the information you want to return.

The **Err** object is an intrinsic object with global scope. There is no need to create an instance of it in your code.

Error Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns the error message that corresponds to a given error number.

Syntax

Error[(*errornumber*)]

The optional *errornumber* argument can be any valid error number. If *errornumber* is a valid error number, but is not defined, **Error** returns the string "Application-defined or object-defined error." If *errornumber* is not valid, an error occurs. If *errornumber* is omitted, the message corresponding to the most recent run-time error is returned. If no run-time error has occurred, or *errornumber* is 0, **Error** returns a zero-length string ("").

Remarks

Examine the property settings of the **Err** object to identify the most recent run-time error. The return value of the **Error** function corresponds to the **Description** property of the **Err** object.

IsError Function

Returns a **Boolean** value indicating whether an expression is an error value.

Syntax

IsError(*expression*)

The required *expression* argument can be any valid expression.

Remarks

Error values are created by converting real numbers to error values using the **CVErr** function. The **IsError** function is used to determine if a numeric expression represents an error. **IsError** returns **True** if the *expression* argument indicates an error; otherwise, it returns **False**.

DDB Function

Returns a Double specifying the depreciation of an asset for a specific time period using the double-declining balance method or some other method you specify.

Syntax

DDB(*cost*, *salvage*, *life*, *period*[, *factor*])

The **DDB** function has these named arguments:

Part	Description
<i>cost</i>	Required. Double specifying initial cost of the asset.
<i>salvage</i>	Required. Double specifying value of the asset at the end of its useful life.
<i>life</i>	Required. Double specifying length of useful life of the asset.
<i>period</i>	Required. Double specifying period for which asset depreciation is calculated.
<i>factor</i>	Optional. Variant specifying rate at which the balance declines. If omitted, 2 (double-declining method) is assumed.

Remarks

The double-declining balance method computes depreciation at an accelerated rate. Depreciation is highest in the first period and decreases in successive periods.

The *life* and *period* arguments must be expressed in the same units. For example, if *life* is given in months, *period* must also be given in months. All arguments must be positive numbers.

The **DDB** function uses the following formula to calculate depreciation for a given period:

Depreciation / *period* = ((*cost* – *salvage*) * *factor*) / *life*

FV Function

Returns a Double specifying the future value of an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax

FV(*rate*, *nper*, *pmt*[, *pv*[, *type*]])

The **FV** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
<i>nper</i>	Required. Integer specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
<i>pmt</i>	Required. Double specifying payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
<i>pv</i>	Optional. Variant specifying present value (or lump sum) of a series of future payments. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An annuity can be a loan (such as a home mortgage) or an investment (such as a monthly savings plan).

The *rate* and *nper* arguments must be calculated using payment periods expressed in the same units. For example, if *rate* is calculated using months, *nper* must also be calculated using months.

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

IPmt Function

Returns a Double specifying the interest payment for a given period of an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax

IPmt(*rate*, *per*, *nper*, *pv*[, *fv*[, *type*]])

The **IPmt** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
<i>per</i>	Required. Double specifying payment period in the range 1 through <i>nper</i> .
<i>nper</i>	Required. Double specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
<i>pv</i>	Required. Double specifying present value, or value today, of a series of future payments or receipts. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make.
<i>fv</i>	Optional. Variant specifying future value or cash balance you want after you've made the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An

annuity can be a loan (such as a home mortgage) or an investment (such as a monthly savings plan).

The *rate* and *nper* arguments must be calculated using payment periods expressed in the same units. For example, if *rate* is calculated using months, *nper* must also be calculated using months.

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

IRR Function

Returns a Double specifying the internal rate of return for a series of periodic cash flows (payments and receipts).

Syntax

IRR(*values*)[, *guess*])

The **IRR** function has these named arguments:

Part	Description
<i>values</i> ()	Required. Array of Double specifying cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt).
<i>guess</i>	Optional. Variant specifying value you estimate will be returned by IRR . If omitted, <i>guess</i> is 0.1 (10 percent).

Remarks

The internal rate of return is the interest rate received for an investment consisting of payments and receipts that occur at regular intervals.

The **IRR** function uses the order of values within the array to interpret the order of payments and receipts. Be sure to enter your payment and receipt values in the correct sequence. The cash flow for each period doesn't have to be fixed, as it is for an annuity.

IRR is calculated by iteration. Starting with the value of *guess*, **IRR** cycles through the calculation until the result is accurate to within 0.00001 percent. If **IRR** can't find a result after 20 tries, it fails.

MIRR Function

Returns a Double specifying the modified internal rate of return for a series of periodic cash flows (payments and receipts).

Syntax

MIRR(*values*()), *finance_rate*, *reinvest_rate*)

The **MIRR** function has these named arguments:

Part	Description
<i>values</i> ()	Required. Array of Double specifying cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt).
<i>finance_rate</i>	Required. Double specifying interest rate paid as the cost of financing.
<i>reinvest_rate</i>	Required. Double specifying interest rate received on gains from cash reinvestment.

Remarks

The modified internal rate of return is the internal rate of return when payments and receipts are financed at different rates. The **MIRR** function takes into account both the cost of the investment (*finance_rate*) and the interest rate received on reinvestment of cash (*reinvest_rate*).

The *finance_rate* and *reinvest_rate* arguments are percentages expressed as decimal values. For example, 12 percent is expressed as 0.12.

The **MIRR** function uses the order of values within the array to interpret the order of payments and receipts. Be sure to enter your payment and receipt values in the correct sequence.

NPer Function

Returns a Double specifying the number of periods for an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax

NPer(*rate*, *pmt*, *pv*[, *fv*[, *type*]])

The **NPer** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
<i>pmt</i>	Required. Double specifying payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
<i>pv</i>	Required. Double specifying present value, or value today, of a series of future payments or receipts. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make.
<i>fv</i>	Optional. Variant specifying future value or cash balance you want after you've made the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An annuity can be a loan (such as a home mortgage) or an investment (such as a

monthly savings plan).

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

Pmt Function

Returns a Double specifying the payment for an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax

Pmt(*rate*, *nper*, *pv*[, *fv*[, *type*]])

The **Pmt** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
<i>nper</i>	Required. Integer specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
<i>pv</i>	Required. Double specifying present value (or lump sum) that a series of payments to be paid in the future is worth now. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make.
<i>fv</i>	Optional. Variant specifying future value or cash balance you want after you've made the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An annuity can be a loan (such as a home mortgage) or an investment (such as a

monthly savings plan).

The ***rate*** and ***nper*** arguments must be calculated using payment periods expressed in the same units. For example, if ***rate*** is calculated using months, ***nper*** must also be calculated using months.

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

PPmt Function

Returns a Double specifying the principal payment for a given period of an annuity based on periodic, fixed payments and a fixed interest rate.

Syntax

PPmt(*rate*, *per*, *nper*, *pv*[, *fv*[, *type*]])

The **PPmt** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
<i>per</i>	Required. Integer specifying payment period in the range 1 through <i>nper</i> .
<i>nper</i>	Required. Integer specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
<i>pv</i>	Required. Double specifying present value, or value today, of a series of future payments or receipts. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make.
<i>fv</i>	Optional. Variant specifying future value or cash balance you want after you've made the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An

annuity can be a loan (such as a home mortgage) or an investment (such as a monthly savings plan).

The *rate* and *nper* arguments must be calculated using payment periods expressed in the same units. For example, if *rate* is calculated using months, *nper* must also be calculated using months.

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

PV Function

Returns a Double specifying the present value of an annuity based on periodic, fixed payments to be paid in the future and a fixed interest rate.

Syntax

PV(*rate*, *nper*, *pmt*[, *fv*[, *type*]])

The **PV** function has these named arguments:

Part	Description
rate	Required. Double specifying interest rate per period. For example, if you get a car loan at an annual percentage rate (APR) of 10 percent and make monthly payments, the rate per period is 0.1/12, or 0.0083.
nper	Required. Integer specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
pmt	Required. Double specifying payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
fv	Optional. Variant specifying future value or cash balance you want after you've made the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
type	Optional. Variant specifying when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.

Remarks

An annuity is a series of fixed cash payments made over a period of time. An annuity can be a loan (such as a home mortgage) or an investment (such as a monthly savings plan).

The *rate* and *nper* arguments must be calculated using payment periods expressed in the same units. For example, if *rate* is calculated using months, *nper* must also be calculated using months.

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

Rate Function

Returns a Double specifying the interest rate per period for an annuity.

Syntax

Rate(*nper*, *pmt*, *pv*[, *fv*[, *type*[, *guess*]]])

The **Rate** function has these named arguments:

Part	Description
<i>nper</i>	Required. Double specifying total number of payment periods in the annuity. For example, if you make monthly payments on a four-year car loan, your loan has a total of 4 * 12 (or 48) payment periods.
<i>pmt</i>	Required. Double specifying payment to be made each period. Payments usually contain principal and interest that doesn't change over the life of the annuity.
<i>pv</i>	Required. Double specifying present value, or value today, of a series of future payments or receipts. For example, when you borrow money to buy a car, the loan amount is the present value to the lender of the monthly car payments you will make.
<i>fv</i>	Optional. Variant specifying future value or cash balance you want after you make the final payment. For example, the future value of a loan is \$0 because that's its value after the final payment. However, if you want to save \$50,000 over 18 years for your child's education, then \$50,000 is the future value. If omitted, 0 is assumed.
<i>type</i>	Optional. Variant specifying a number indicating when payments are due. Use 0 if payments are due at the end of the payment period, or use 1 if payments are due at the beginning of the period. If omitted, 0 is assumed.
<i>guess</i>	Optional. Variant specifying value you estimate will be returned by Rate . If omitted, <i>guess</i> is 0.1 (10 percent).

Remarks

An annuity is a series of fixed cash payments made over a period of time. An annuity can be a loan (such as a home mortgage) or an investment (such as a monthly savings plan).

For all arguments, cash paid out (such as deposits to savings) is represented by negative numbers; cash received (such as dividend checks) is represented by positive numbers.

Rate is calculated by iteration. Starting with the value of *guess*, **Rate** cycles through the calculation until the result is accurate to within 0.00001 percent. If **Rate** can't find a result after 20 tries, it fails. If your guess is 10 percent and **Rate** fails, try a different value for *guess*.

SLN Function

Returns a Double specifying the straight-line depreciation of an asset for a single period.

Syntax

SLN(*cost*, *salvage*, *life*)

The **SLN** function has these named arguments:

Part	Description
<i>cost</i>	Required. Double specifying initial cost of the asset.
<i>salvage</i>	Required. Double specifying value of the asset at the end of its useful life.
<i>life</i>	Required. Double specifying length of the useful life of the asset.

Remarks

The depreciation period must be expressed in the same unit as the *life* argument. All arguments must be positive numbers.

SYD Function

Returns a Double specifying the sum-of-years' digits depreciation of an asset for a specified period.

Syntax

SYD(*cost*, *salvage*, *life*, *period*)

The **SYD** function has these named arguments:

Part	Description
<i>cost</i>	Required. Double specifying initial cost of the asset.
<i>salvage</i>	Required. Double specifying value of the asset at the end of its useful life.
<i>life</i>	Required. Double specifying length of the useful life of the asset.
<i>period</i>	Required. Double specifying period for which asset depreciation is calculated.

Remarks

The *life* and *period* arguments must be expressed in the same units. For example, if *life* is given in months, *period* must also be given in months. All arguments must be positive numbers.

Command Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns the argument portion of the command line used to launch Microsoft Visual Basic or an executable program developed with Visual Basic. The Visual Basic **Command** function is not available in Microsoft Office applications.

Syntax

Command

Remarks

When Visual Basic is launched from the command line, any portion of the command line that follows `/cmd` is passed to the program as the command-line argument. In the following example, `cmdlineargs` represents the argument information returned by the **Command** function.

```
VB /cmd cmdlineargs
```

For applications developed with Visual Basic and compiled to an `.exe` file, **Command** returns any arguments that appear after the name of the application on the command line. For example:

```
MyApp cmdlineargs
```

To find how command line arguments can be changed in the user interface of the application you're using, search Help for "command line arguments."

DeleteSetting Statement

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Deletes a section or key setting from an application's entry in the Windows registry or (on the Macintosh) information in the application's initialization file.

Syntax

DeleteSetting *appname*, *section*[, *key*]

The **DeleteSetting** statement syntax has these named arguments:

Part	Description
<i>appname</i>	Required. String expression containing the name of the application or project to which the section or key setting applies. On the Macintosh, this is the filename of the initialization file in the Preferences folder in the System folder.
<i>section</i>	Required. String expression containing the name of the section where the key setting is being deleted. If only <i>appname</i> and <i>section</i> are provided, the specified section is deleted along with all related key settings.
<i>key</i>	Optional. String expression containing the name of the key setting being deleted.

Remarks

If all arguments are provided, the specified setting is deleted. A run-time error occurs if you attempt to use the **DeleteSetting** statement on a non-existent section or key setting.

DoEvents Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Yields execution so that the operating system can process other events.

Syntax

DoEvents()

Remarks

The **DoEvents** function returns an Integer representing the number of open forms in stand-alone versions of Visual Basic, such as Visual Basic, Professional Edition. **DoEvents** returns zero in all other applications.

DoEvents passes control to the operating system. Control is returned after the operating system has finished processing the events in its queue and all keys in the **SendKeys** queue have been sent.

DoEvents is most useful for simple things like allowing a user to cancel a process after it has started, for example a search for a file. For long-running processes, yielding the processor is better accomplished by using a Timer or delegating the task to an ActiveX EXE component.. In the latter case, the task can continue completely independent of your application, and the operating system takes care of multitasking and time slicing.

Caution Any time you temporarily yield the processor within an event procedure, make sure the procedure is not executed again from a different part of your code before the first call returns; this could cause unpredictable results. In addition, do not use **DoEvents** if other applications could possibly interact with your procedure in unforeseen ways during the time you have yielded control.

Environ Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns the **String** associated with an operating system environment variable. Not available on the Macintosh

Syntax

Environ({*envstring* | *number*})

The **Environ** function syntax has these named arguments:

Part	Description
<i>envstring</i>	Optional. String expression containing the name of an environment variable.
<i>number</i>	Optional. Numeric expression corresponding to the numeric order of the environment string in the environment-string table. The <i>number</i> argument can be any numeric expression, but is rounded to a whole number before it is evaluated.

Remarks

If *envstring* can't be found in the environment-string table, a zero-length string ("") is returned. Otherwise, **Environ** returns the text assigned to the specified *envstring*; that is, the text following the equal sign (=) in the environment-string table for that environment variable.

If you specify *number*, the string occupying that numeric position in the environment-string table is returned. In this case, **Environ** returns all of the text, including *envstring*. If there is no environment string in the specified position, **Environ** returns a zero-length string.

FileDateTime Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a **Variant (Date)** that indicates the date and time when a file was created or last modified.

Syntax

FileDateTime(*pathname*)

The required *pathname* argument is a string expression that specifies a file name. The *pathname* may include the directory or folder, and the drive.

FileLen Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a Long specifying the length of a file in bytes.

Syntax

FileLen(*pathname*)

The required *pathname* argument is a string expression that specifies a file. The *pathname* may include the directory or folder, and the drive.

Remarks

If the specified file is open when the **FileLen** function is called, the value returned represents the size of the file immediately before it was opened.

Note To obtain the length of an open file, use the **LOF** function.

FreeFile Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns an Integer representing the next file number available for use by the **Open** statement.

Syntax

FreeFile[(*rangenum*)]

The optional *rangenum* argument is a Variant that specifies the range from which the next free file number is to be returned. Specify a 0 (default) to return a file number in the range 1 – 255, inclusive. Specify a 1 to return a file number in the range 256 – 511.

Remarks

Use **FreeFile** to supply a file number that is not already in use.

GetAllSettings Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a list of key settings and their respective values (originally created with **SaveSetting**) from an application's entry in the Windows registry or (on the Macintosh) information in the application's initialization file.

Syntax

GetAllSettings(*appname*, *section*)

The **GetAllSettings** function syntax has these named arguments:

Part	Description
<i>appname</i>	Required. containing the String expression name of the application or project whose key settings are requested. On the Macintosh, this is the filename of the initialization file in the Preferences folder in the System folder.
<i>section</i>	Required. String expression containing the name of the section whose key settings are requested. GetAllSettings returns a Variant whose contents is a two-dimensional array of strings containing all the key settings in the specified section and their corresponding values.

Remarks

GetAllSettings returns an uninitialized **Variant** if either *appname* or *section* does not exist.

GetSetting Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a key setting value from an application's entry in the Windows registry or (on the Macintosh) information in the application's initialization file.

Syntax

GetSetting(*appname*, *section*, *key*[, *default*])

The **GetSetting** function syntax has these named arguments:

Part	Description
<i>appname</i>	Required. String expression containing the name of the application or project whose key setting is requested. On the Macintosh, this is the filename of the initialization file in the Preferences folder in the System folder.
<i>section</i>	Required. String expression containing the name of the section where the key setting is found.
<i>key</i>	Required. String expression containing the name of the key setting to return.
<i>default</i>	Optional. Expression containing the value to return if no value is set in the key setting. If omitted, default is assumed to be a zero-length string ("").

Remarks

If any of the items named in the **GetSetting** arguments do not exist, **GetSetting** returns the value of **default**.

QBColor Function

Returns a Long representing the RGB color code corresponding to the specified color number.

Syntax

QBColor(*color*)

The required *color* argument is a whole number in the range 0–15.

Settings

The *color* argument has these settings:

Number	Color	Number	Color
0	Black	8	Gray
1	Blue	9	Light Blue
2	Green	10	Light Green
3	Cyan	11	Light Cyan
4	Red	12	Light Red
5	Magenta	13	Light Magenta
6	Yellow	14	Light Yellow
7	White	15	Bright White

Remarks

The *color* argument represents color values used by earlier versions of Basic (such as Microsoft Visual Basic for MS-DOS and the Basic Compiler). Starting with the least-significant byte, the returned value specifies the red, green, and blue values used to set the appropriate color in the RGB system used by Visual Basic for Applications.

RGB Function

Returns a Long whole number representing an RGB color value.

Syntax

RGB(*red*, *green*, *blue*)

The **RGB** function syntax has these named arguments:

Part	Description
<i>red</i>	Required; Variant (Integer) . Number in the range 0–255, inclusive, that represents the red component of the color.
<i>green</i>	Required; Variant (Integer) . Number in the range 0–255, inclusive, that represents the green component of the color.
<i>blue</i>	Required; Variant (Integer) . Number in the range 0–255, inclusive, that represents the blue component of the color.

Remarks

Application methods and properties that accept a color specification expect that specification to be a number representing an RGB color value. An RGB color value specifies the relative intensity of red, green, and blue to cause a specific color to be displayed.

The value for any argument to **RGB** that exceeds 255 is assumed to be 255.

The following table lists some standard colors and the red, green, and blue values they include:

Color	Red Value	Green Value	Blue Value
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan	0	255	255

Red	255	0	0
Magenta	255	0	255
Yellow	255	255	0
White	255	255	255

The RGB color values returned by this function are incompatible with those used by the Macintosh operating system. They may be used within the context of Microsoft applications for the Macintosh, but should not be used when communicating color changes directly to the Macintosh operating system.

SaveSetting Statement

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Saves or creates an application entry in the application's entry in the Windows registry or (on the Macintosh) information in the application's initialization file.

Syntax

SaveSetting *appname, section, key, setting*

The **SaveSetting** statement syntax has these named arguments:

Part	Description
<i>appname</i>	Required. String expression containing the name of the application or project to which the setting applies. On the Macintosh, this is the filename of the initialization file in the Preferences folder in the System folder.
<i>section</i>	Required. String expression containing the name of the section where the key setting is being saved.
<i>key</i>	Required. String expression containing the name of the key setting being saved.
<i>setting</i>	Required. Expression containing the value that <i>key</i> is being set to.

Remarks

An error occurs if the key setting can't be saved for any reason.

Shell Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Runs an executable program and returns a **Variant (Double)** representing the program's task ID if successful, otherwise it returns zero.

Syntax

Shell(*pathname*[,*windowstyle*])

The **Shell** function syntax has these named arguments:

Part	Description
<i>pathname</i>	Required; Variant (String) . Name of the program to execute and any required arguments or command-line switches; may include directory or folder and drive. On the Macintosh, you can use the MacID function to specify an application's signature instead of its name. The following example uses the signature for Microsoft Word: Shell MacID("MSWD")
<i>windowstyle</i>	Optional. Variant (Integer) corresponding to the style of the window in which the program is to be run. If <i>windowstyle</i> is omitted, the program is started minimized with focus. On the Macintosh (System 7.0 or later), <i>windowstyle</i> only determines whether or not the application gets the focus when it is run.

The *windowstyle* named argument has these values:

Constant	Value	Description
vbHide	0	Window is hidden and focus is passed to the hidden window. The vbHide constant is not

		applicable on Macintosh platforms.
vbNormalFocus	1	Window has focus and is restored to its original size and position.
vbMinimizedFocus	2	Window is displayed as an icon with focus.
vbMaximizedFocus	3	Window is maximized with focus.
vbNormalNoFocus	4	Window is restored to its most recent size and position. The currently active window remains active.
vbMinimizedNoFocus	6	Window is displayed as an icon. The currently active window remains active.

Remarks

If the **Shell** function successfully executes the named file, it returns the task ID of the started program. The task ID is a unique number that identifies the running program. If the **Shell** function can't start the named program, an error occurs.

On the Macintosh, **vbNormalFocus**, **vbMinimizedFocus**, and **vbMaximizedFocus** all place the application in the foreground; **vbHide**, **vbNoFocus**, **vbMinimizeFocus** all place the application in the background.

Note By default, the **Shell** function runs other programs asynchronously. This means that a program started with **Shell** might not finish executing before the statements following the **Shell** function are executed.

NPV Function

Returns a Double specifying the net present value of an investment based on a series of periodic cash flows (payments and receipts) and a discount rate.

Syntax

NPV(*rate*, *values*())

The **NPV** function has these named arguments:

Part	Description
<i>rate</i>	Required. Double specifying discount rate over the length of the period, expressed as a decimal.
<i>values</i> ()	Required. Array of Double specifying cash flow values. The array must contain at least one negative value (a payment) and one positive value (a receipt).

Remarks

The net present value of an investment is the current value of a future series of payments and receipts.

The **NPV** function uses the order of values within the array to interpret the order of payments and receipts. Be sure to enter your payment and receipt values in the correct sequence.

The **NPV** investment begins one period before the date of the first cash flow value and ends with the last cash flow value in the array.

The net present value calculation is based on future cash flows. If your first cash flow occurs at the beginning of the first period, the first value must be added to the value returned by **NPV** and must not be included in the cash flow values of *values*().

The **NPV** function is similar to the **PV** function (present value) except that the

PV function allows cash flows to begin either at the end or the beginning of a period. Unlike the variable **NPV** cash flow values, **PV** cash flows must be fixed throughout the investment.

Version Property

Returns a **String** indicating the version number of the currently installed copy of Microsoft Access. Read-only.

expression.**Version**

expression Required. An expression that returns one of the objects in the Applies To list.

CurDir Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a **Variant (String)** representing the current path.

Syntax

CurDir[(*drive*)]

The optional *drive* argument is a string expression that specifies an existing drive. If no drive is specified or if *drive* is a zero-length string (""), **CurDir** returns the path for the current drive. On the Macintosh, **CurDir** ignores any *drive* specified and simply returns the path for the current drive.

Dir Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a **String** representing the name of a file, directory, or folder that matches a specified pattern or file attribute, or the volume label of a drive.

Syntax

Dir[(*pathname*[, *attributes*])]

The **Dir** function syntax has these parts:

Part	Description
<i>pathname</i>	Optional. String expression that specifies a file name— may include directory or folder, and drive. A zero-length string ("") is returned if <i>pathname</i> is not found.
<i>attributes</i>	Optional. Constant or numeric expression, whose sum specifies file attributes. If omitted, returns files that match <i>pathname</i> but have no attributes.

Settings

The *attributes* argument settings are:

Constant	Value	Description
vbNormal	0	(Default) Specifies files with no attributes.
vbReadOnly	1	Specifies read-only files in addition to files with no attributes.
vbHidden	2	Specifies hidden files in addition to files with no attributes.
VbSystem	4	Specifies system files in addition to files with no attributes. Not available on the Macintosh.

vbVolume	8	Specifies volume label; if any other attributed is specified, vbVolume is ignored. Not available on the Macintosh.
vbDirectory	16	Specifies directories or folders in addition to files with no attributes.
vbAlias	64	Specified file name is an alias. Available only on the Macintosh.

Note These constants are specified by Visual Basic for Applications and can be used anywhere in your code in place of the actual values.

Remarks

In Microsoft Windows, **Dir** supports the use of multiple character (*) and single character (?) wildcards to specify multiple files. On the Macintosh, these characters are treated as valid file name characters and can't be used as wildcards to specify multiple files.

Since the Macintosh doesn't support the wildcards, use the file type to identify groups of files. You can use the **MacID** function to specify file type instead of using the file names. For example, the following statement returns the name of the first TEXT file in the current folder:

```
Dir("SomePath", MacID("TEXT"))
```

To iterate over all files in a folder, specify an empty string:

```
Dir("")
```

If you use the **MacID** function with **Dir** in Microsoft Windows, an error occurs.

Any *attribute* value greater than 256 is considered a **MacID** value.

You must specify *pathname* the first time you call the **Dir** function, or an error occurs. If you also specify file attributes, *pathname* must be included.

Dir returns the first file name that matches *pathname*. To get any additional file names that match *pathname*, call **Dir** again with no arguments. When no more

file names match, **Dir** returns a zero-length string (""). Once a zero-length string is returned, you must specify *pathname* in subsequent calls or an error occurs. You can change to a new *pathname* without retrieving all of the file names that match the current *pathname*. However, you can't call the **Dir** function recursively. Calling **Dir** with the **vbDirectory** attribute does not continually return subdirectories.

Tip Because file names are retrieved in no particular order, you may want to store returned file names in an array, and then sort the array.

EOF Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns an Integer containing the Boolean value **True** when the end of a file opened for **Random** or sequential **Input** has been reached.

Syntax

EOF(*filenumber*)

The required *filenumber* argument is an **Integer** containing any valid file number.

Remarks

Use **EOF** to avoid the error generated by attempting to get input past the end of a file.

The **EOF** function returns **False** until the end of the file has been reached. With files opened for **Random** or **Binary** access, **EOF** returns **False** until the last executed **Get** statement is unable to read an entire record.

With files opened for **Binary** access, an attempt to read through the file using the **Input** function until **EOF** returns **True** generates an error. Use the **LOF** and **Loc** functions instead of **EOF** when reading binary files with **Input**, or use **Get** when using the **EOF** function. With files opened for **Output**, **EOF** always returns **True**.

FileAttr Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a Long representing the file mode for files opened using the **Open** statement.

Syntax

FileAttr(*filenumber*, *returntype*)

The **FileAttr** function syntax has these named arguments:

Part	Description
<i>filenumber</i>	Required; Integer. Any valid file number.
<i>returntype</i>	Required; Integer . Number indicating the type of information to return. Specify 1 to return a value indicating the file mode. On 16-bit systems only, specify 2 to retrieve an operating system file handle. Returntype 2 is not supported in 32-bit systems and causes an error.

Return Values

When the *returntype* argument is 1, the following return values indicate the file access mode:

Mode	Value
Input	1
Output	2
Random	4
Append	8
Binary	32

GetAttr Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns an **Integer** representing the attributes of a file, directory, or folder.

Syntax

GetAttr(*pathname*)

The required *pathname* argument is a string expression that specifies a file name. The *pathname* may include the directory or folder, and the drive.

Return Values

The value returned by **GetAttr** is the sum of the following attribute values:

Constant	Value	Description
vbNormal	0	Normal.
vbReadOnly	1	Read-only.
vbHidden	2	Hidden.
vbSystem	4	System file. Not available on the Macintosh.
vbDirectory	16	Directory or folder.
vbArchive	32	File has changed since last backup. Not available on the Macintosh.
vbAlias	64	Specified file name is an alias. Available only on the Macintosh.

Note These constants are specified by Visual Basic for Applications. The names can be used anywhere in your code in place of the actual values.

Remarks

To determine which attributes are set, use the **And** operator to perform a bitwise comparison of the value returned by the **GetAttr** function and the value of the individual file attribute you want. If the result is not zero, that attribute is set for the named file. For example, the return value of the following **And** expression is zero if the Archive attribute is not set:

```
Result = GetAttr(FName) And vbArchive
```

A nonzero value is returned if the Archive attribute is set.

Input Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns String containing characters from a file opened in **Input** or **Binary** mode.

Syntax

Input(*number*, [#]*filenumber*)

The **Input** function syntax has these parts:

Part	Description
<i>number</i>	Required. Any valid numeric expression specifying the number of characters to return.
<i>filenumber</i>	Required. Any valid file number.

Remarks

Data read with the **Input** function is usually written to a file with **Print #** or **Put**. Use this function only with files opened in **Input** or **Binary** mode.

Unlike the **Input #** statement, the **Input** function returns all of the characters it reads, including commas, carriage returns, linefeeds, quotation marks, and leading spaces.

With files opened for **Binary** access, an attempt to read through the file using the **Input** function until **EOF** returns **True** generates an error. Use the **LOF** and **Loc** functions instead of **EOF** when reading binary files with **Input**, or use **Get** when using the **EOF** function.

Note Use the **InputB** function for byte data contained within text files. With

InputB, *number* specifies the number of bytes to return rather than the number of characters to return.

Loc Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a Long specifying the current read/write position within an open file.

Syntax

Loc(*filenumber*)

The required *filenumber* argument is any valid Integer file number.

Remarks

The following describes the return value for each file access mode:

Mode	Return Value
Random	Number of the last record read from or written to the file.
Sequential	Current byte position in the file divided by 128. However, information returned by Loc for sequential files is neither used nor required.
Binary	Position of the last byte read or written.

LOF Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a Long representing the size, in bytes, of a file opened using the **Open** statement.

Syntax

LOF(*filenumber*)

The required *filenumber* argument is an Integer containing a valid file number.

Note Use the **FileLen** function to obtain the length of a file that is not open.

Seek Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Returns a Long specifying the current read/write position within a file opened using the **Open** statement.

Syntax

Seek(*filenumber*)

The required *filenumber* argument is an Integer containing a valid file number.

Remarks

Seek returns a value between 1 and 2,147,483,647 (equivalent to $2^{31} - 1$), inclusive.

The following describes the return values for each file access mode.

Mode	Return Value
Random	Number of the next record read or written
Binary , Output , Append , Input	Byte position at which the next operation takes place. The first byte in a file is at position 1, the second byte is at position 2, and so on.

SetAttr Statement

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Sets attribute information for a file.

Syntax

SetAttr *pathname*, *attributes*

The **SetAttr** statement syntax has these named arguments:

Part	Description
<i>pathname</i>	Required. String expression that specifies a file name— may include directory or folder, and drive.
<i>attributes</i>	Required. Constant or numeric expression, whose sum specifies file attributes.

Settings

The *attributes* argument settings are:

Constant	Value	Description
vbNormal	0	Normal (default).
vbReadOnly	1	Read-only.
vbHidden	2	Hidden.
vbSystem	4	System file. Not available on the Macintosh.
vbArchive	32	File has changed since last backup.
vbAlias	64	Specified file name is an alias. Available only on the Macintosh.

Note These constants are specified by Visual Basic for Applications. The names can be used anywhere in your code in place of the actual values.

Remarks

A run-time error occurs if you try to set the attributes of an open file.

Spc Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Used with the **Print #** statement or the **Print** method to position output.

Syntax

Spc(*n*)

The required *n* argument is the number of spaces to insert before displaying or printing the next expression in a list.

Remarks

If *n* is less than the output line width, the next print position immediately follows the number of spaces printed. If *n* is greater than the output line width, **Spc** calculates the next print position using the formula:

$$\text{currentprintposition} + (n \text{ Mod } \text{width})$$

For example, if the current print position is 24, the output line width is 80, and you specify **Spc(90)**, the next print will start at position 34 (current print position + the remainder of 90/80). If the difference between the current print position and the output line width is less than *n* (or *n Mod width*), the **Spc** function skips to the beginning of the next line and generates spaces equal to $n - (\text{width} - \text{currentprintposition})$.

Note Make sure your tabular columns are wide enough to accommodate wide letters.

When you use the **Print** method with a proportionally spaced font, the width of space characters printed using the **Spc** function is always an average of the width of all characters in the point size for the chosen font. However, there is no correlation between the number of characters printed and the number of fixed-

width columns those characters occupy. For example, the uppercase letter W occupies more than one fixed-width column and the lowercase letter i occupies less than one fixed-width column.

Tab Function

Note The function, method, object, or property described in this topic is disabled if the Microsoft Jet Expression Service is running in sandbox mode, which prevents the evaluation of potentially unsafe expressions. For more information on sandbox mode, search for "sandbox mode" in Help.

Used with the **Print #** statement or the **Print** method to position output.

Syntax

Tab[(*n*)]

The optional *n* argument is the column number moved to before displaying or printing the next expression in a list. If omitted, **Tab** moves the insertion point to the beginning of the next print zone. This allows **Tab** to be used instead of a comma in locales where the comma is used as a decimal separator.

Remarks

If the current print position on the current line is greater than *n*, **Tab** skips to the *n*th column on the next output line. If *n* is less than 1, **Tab** moves the print position to column 1. If *n* is greater than the output line width, **Tab** calculates the next print position using the formula:

n Mod width

For example, if *width* is 80 and you specify **Tab**(90), the next print will start at column 10 (the remainder of 90/80). If *n* is less than the current print position, printing begins on the next line at the calculated print position. If the calculated print position is greater than the current print position, printing begins at the calculated print position on the same line.

The leftmost print position on an output line is always 1. When you use the **Print #** statement to print to files, the rightmost print position is the current width of the output file, which you can set using the **Width #** statement.

Note Make sure your tabular columns are wide enough to accommodate wide

letters.

When you use the **Tab** function with the **Print** method, the print surface is divided into uniform, fixed-width columns. The width of each column is an average of the width of all characters in the point size for the chosen font. However, there is no correlation between the number of characters printed and the number of fixed-width columns those characters occupy. For example, the uppercase letter W occupies more than one fixed-width column and the lowercase letter i occupies less than one fixed-width column.

IsArray Function

Returns a **Boolean** value indicating whether a variable is an array.

Syntax

IsArray(*varname*)

The required *varname* argument is an identifier specifying a variable.

Remarks

IsArray returns **True** if the variable is an array; otherwise, it returns **False**. **IsArray** is especially useful with variants containing arrays.

IsEmpty Function

Returns a **Boolean** value indicating whether a variable has been initialized.

Syntax

IsEmpty(*expression*)

The required *expression* argument is a Variant containing a numeric or string expression. However, because **IsEmpty** is used to determine if individual variables are initialized, the *expression* argument is most often a single variable name.

Remarks

IsEmpty returns **True** if the variable is uninitialized, or is explicitly set to Empty; otherwise, it returns **False**. **False** is always returned if *expression* contains more than one variable. **IsEmpty** only returns meaningful information for variants.

IsMissing Function

Returns a **Boolean** value indicating whether an optional **Variant** argument has been passed to a procedure.

Syntax

IsMissing(*argname*)

The required *argname* argument contains the name of an optional **Variant** procedure argument.

Remarks

Use the **IsMissing** function to detect whether or not optional **Variant** arguments have been provided in calling a procedure. **IsMissing** returns **True** if no value has been passed for the specified argument; otherwise, it returns **False**. If **IsMissing** returns **True** for an argument, use of the missing argument in other code may cause a user-defined error. If **IsMissing** is used on a **ParamArray** argument, it always returns **False**. To detect an empty **ParamArray**, test to see if the array's upper bound is less than its lower bound.

Note **IsMissing** does not work on simple data types (such as **Integer** or **Double**) because, unlike **Variants**, they don't have a provision for a "missing" flag bit. Because of this, the syntax for typed optional arguments allows you to specify a default value. If the argument is omitted when the procedure is called, then the argument will have this default value, as in the example below:

```
Sub MySub(Optional MyVar As String = "specialvalue")
    If MyVar = "specialvalue" Then
        ' MyVar was omitted.
    Else
        ...
    End Sub
```

In many cases you can omit the `If MyVar` test entirely by making the default value equal to the value you want `MyVar` to contain if the user omits it from the function call. This makes your code more concise and efficient.

IsNull Function

Returns a **Boolean** value that indicates whether an expression contains no valid data (Null).

Syntax

IsNull(*expression*)

The required *expression* argument is a Variant containing a numeric expression or string expression.

Remarks

IsNull returns **True** if *expression* is **Null**; otherwise, **IsNull** returns **False**. If *expression* consists of more than one variable, **Null** in any constituent variable causes **True** to be returned for the entire expression.

The **Null** value indicates that the **Variant** contains no valid data. **Null** is not the same as Empty, which indicates that a variable has not yet been initialized. It is also not the same as a zero-length string (""), which is sometimes referred to as a null string.

Important Use the **IsNull** function to determine whether an expression contains a **Null** value. Expressions that you might expect to evaluate to **True** under some circumstances, such as `If Var = Null` and `If Var <> Null`, are always **False**. This is because any expression containing a **Null** is itself **Null** and, therefore, **False**.

IsNumeric Function

Returns a **Boolean** value indicating whether an expression can be evaluated as a number.

Syntax

IsNumeric(*expression*)

The required *expression* argument is a Variant containing a numeric expression or string expression.

Remarks

IsNumeric returns **True** if the entire *expression* is recognized as a number; otherwise, it returns **False**.

IsNumeric returns **False** if *expression* is a date expression.

IsObject Function

Returns a **Boolean** value indicating whether an identifier represents an object variable.

Syntax

IsObject(*identifier*)

The required *identifier* argument is a variable name.

Remarks

IsObject is useful only in determining whether a Variant is of **VarType vbObject**. This could occur if the **Variant** actually references (or once referenced) an object, or if it contains **Nothing**.

IsObject returns **True** if *identifier* is a variable declared with Object type or any valid class type, or if *identifier* is a **Variant** of **VarType vbObject**, or a user-defined object; otherwise, it returns **False**. **IsObject** returns **True** even if the variable has been set to **Nothing**.

Use error trapping to be sure that an object reference is valid.

TypeName Function

Returns a **String** that provides information about a variable.

Syntax

TypeName(*varname*)

The required *varname* argument is a Variant containing any variable except a variable of a user-defined type.

Remarks

The string returned by **TypeName** can be any one of the following:

String returned	Variable
object type	An object whose type is <i>objecttype</i>
Byte	Byte value
Integer	Integer
Long	Long integer
Single	Single-precision floating-point number
Double	Double-precision floating-point number
Currency	Currency value
Decimal	Decimal value
Date	Date value
String	String
Boolean	Boolean value
Error	An error value
Empty	Uninitialized
Null	No valid data
Object	An object
Unknown	An object whose type is unknown
Nothing	Object variable that doesn't refer to an object

If *varname* is an array, the returned string can be any one of the possible returned strings (or **Variant**) with empty parentheses appended. For example, if *varname* is an array of integers, **TypeName** returns "Integer()".

VarType Function

Returns an **Integer** indicating the subtype of a variable.

Syntax

VarType(*varname*)

The required *varname* argument is a Variant containing any variable except a variable of a user-defined type.

Return Values

Constant	Value	Description
vbEmpty	0	Empty (uninitialized)
vbNull	1	Null (no valid data)
vbInteger	2	Integer
vbLong	3	Long integer
vbSingle	4	Single-precision floating-point number
vbDouble	5	Double-precision floating-point number
vbCurrency	6	Currency value
vbDate	7	Date value
vbString	8	String
vbObject	9	Object
vbError	10	Error value
vbBoolean	11	Boolean value
vbVariant	12	Variant (used only with arrays of variants)
vbDataObject	13	A data access object
vbDecimal	14	Decimal value
vbByte	17	Byte value
vbUserDefinedType	36	Variants that contain user-defined types
vbArray	8192	Array

Note These constants are specified by Visual Basic for Applications. The names can be used anywhere in your code in place of the actual values.

Remarks

The **VarType** function never returns the value for **vbArray** by itself. It is always added to some other value to indicate an array of a particular type. The constant **vbVariant** is only returned in conjunction with **vbArray** to indicate that the argument to the **VarType** function is an array of type **Variant**. For example, the value returned for an array of integers is calculated as **vbInteger + vbArray**, or 8194. If an object has a default property, **VarType (object)** returns the type of the object's default property.

Abs Function

Returns a value of the same type that is passed to it specifying the absolute value of a number.

Syntax

Abs(*number*)

The required *number* argument can be any valid numeric expression. If *number* contains Null, **Null** is returned; if it is an uninitialized variable, zero is returned.

Remarks

The absolute value of a number is its unsigned magnitude. For example, ABS(-1) and ABS(1) both return 1.

Atn Function

Returns a **Double** specifying the arctangent of a number.

Syntax

Atn(*number*)

The required *number* argument is a Double or any valid numeric expression.

Remarks

The **Atn** function takes the ratio of two sides of a right triangle (*number*) and returns the corresponding angle in radians. The ratio is the length of the side opposite the angle divided by the length of the side adjacent to the angle.

The range of the result is $-\pi/2$ to $\pi/2$ radians.

To convert degrees to radians, multiply degrees by $\pi/180$. To convert radians to degrees, multiply radians by $180/\pi$.

Note **Atn** is the inverse trigonometric function of **Tan**, which takes an angle as its argument and returns the ratio of two sides of a right triangle. Do not confuse **Atn** with the cotangent, which is the simple inverse of a tangent ($1/\text{tangent}$).

Cos Function

Returns a **Double** specifying the cosine of an angle.

Syntax

Cos(*number*)

The required *number* argument is a Double or any valid numeric expression that expresses an angle in radians.

Remarks

The **Cos** function takes an angle and returns the ratio of two sides of a right triangle. The ratio is the length of the side adjacent to the angle divided by the length of the hypotenuse.

The result lies in the range -1 to 1.

To convert degrees to radians, multiply degrees by $\pi / 180$. To convert radians to degrees, multiply radians by $180/\pi$.

Exp Function

Returns a **Double** specifying e (the base of natural logarithms) raised to a power.

Syntax

Exp(*number*)

The required *number* argument is a Double or any valid numeric expression.

Remarks

If the value of *number* exceeds 709.782712893, an error occurs. The constant e is approximately 2.718282.

Note The **Exp** function complements the action of the **Log** function and is sometimes referred to as the antilogarithm.

Int, Fix Functions

Returns the integer portion of a number.

Syntax

Int(*number*)

Fix(*number*)

The required *number* argument is a Double or any valid numeric expression. If *number* contains Null, **Null** is returned.

Remarks

Both **Int** and **Fix** remove the fractional part of *number* and return the resulting integer value.

The difference between **Int** and **Fix** is that if *number* is negative, **Int** returns the first negative integer less than or equal to *number*, whereas **Fix** returns the first negative integer greater than or equal to *number*. For example, **Int** converts -8.4 to -9, and **Fix** converts -8.4 to -8.

Fix(*number*) is equivalent to:

$\text{Sgn}(\textit{number}) * \text{Int}(\text{Abs}(\textit{number}))$

Log Function

Returns a **Double** specifying the natural logarithm of a number.

Syntax

Log(*number*)

The required *number* argument is a Double or any valid numeric expression greater than zero.

Remarks

The natural logarithm is the logarithm to the base *e*. The constant *e* is approximately 2.718282.

You can calculate base-*n* logarithms for any number *x* by dividing the natural logarithm of *x* by the natural logarithm of *n* as follows:

$$\text{Log}_n(x) = \mathbf{Log}(x) / \mathbf{Log}(n)$$

The following example illustrates a custom **Function** that calculates base-10 logarithms:

```
Static Function Log10(X)
    Log10 = Log(X) / Log(10#)
End Function
```

Rnd Function

Returns a **Single** containing a random number.

Syntax

Rnd[(*number*)]

The optional *number* argument is a Single or any valid numeric expression.

Return Values

If <i>number</i> is	Rnd generates
Less than zero	The same number every time, using <i>number</i> as the seed.
Greater than zero	The next random number in the sequence.
Equal to zero	The most recently generated number.
Not supplied	The next random number in the sequence.

Remarks

The **Rnd** function returns a value less than 1 but greater than or equal to zero.

The value of *number* determines how **Rnd** generates a random number:

For any given initial seed, the same number sequence is generated because each successive call to the **Rnd** function uses the previous number as a seed for the next number in the sequence.

Before calling **Rnd**, use the **Randomize** statement without an argument to initialize the random-number generator with a seed based on the system timer.

To produce random integers in a given range, use this formula:

$\text{Int}((\text{upperbound} - \text{lowerbound} + 1) * \text{Rnd} + \text{lowerbound})$

Here, *upperbound* is the highest number in the range, and *lowerbound* is the lowest number in the range.

Note To repeat sequences of random numbers, call **Rnd** with a negative argument immediately before using **Randomize** with a numeric argument. Using **Randomize** with the same value for *number* does not repeat the previous sequence.

Round Function

Description

Returns a number rounded to a specified number of decimal places.

Syntax

Round(*expression* [,*numdecimalplaces*])

The **Round** function syntax has these parts:

Part	Description
<i>expression</i>	Required. Numeric expression being rounded.
<i>numdecimalplaces</i>	Optional. Number indicating how many places to the right of the decimal are included in the rounding. If omitted, integers are returned by the Round function.

Sgn Function

Returns a **Variant (Integer)** indicating the sign of a number.

Syntax

Sgn(*number*)

The required *number* argument can be any valid numeric expression.

Return Values

If <i>number</i> is	Sgn returns
Greater than zero	1
Equal to zero	0
Less than zero	-1

Remarks

The sign of the *number* argument determines the return value of the **Sgn** function.

Sin Function

Returns a **Double** specifying the sine of an angle.

Syntax

Sin(*number*)

The required *number* argument is a Double or any valid numeric expression that expresses an angle in radians.

Remarks

The **Sin** function takes an angle and returns the ratio of two sides of a right triangle. The ratio is the length of the side opposite the angle divided by the length of the hypotenuse.

The result lies in the range -1 to 1.

To convert degrees to radians, multiply degrees by $\pi/180$. To convert radians to degrees, multiply radians by $180/\pi$.

Sqr Function

Returns a **Double** specifying the square root of a number.

Syntax

Sqr(*number*)

The required *number* argument is a Double or any valid numeric expression greater than or equal to zero.

Tan Function

Returns a **Double** specifying the tangent of an angle.

Syntax

Tan(*number*)

The required *number* argument is a Double or any valid numeric expression that expresses an angle in radians.

Remarks

Tan takes an angle and returns the ratio of two sides of a right triangle. The ratio is the length of the side opposite the angle divided by the length of the side adjacent to the angle.

To convert degrees to radians, multiply degrees by $\pi/180$. To convert radians to degrees, multiply radians by $180/\pi$.

InputDialog Function

Displays a prompt in a dialog box, waits for the user to input text or click a button, and returns a String containing the contents of the text box.

Syntax

InputDialog(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

The **InputDialog** function syntax has these named arguments:

Part	Description
prompt	Required. String expression displayed as the message in the dialog box. The maximum length of prompt is approximately 1024 characters, depending on the width of the characters used. If prompt consists of more than one line, you can separate the lines using a carriage return character (Chr(13)), a linefeed character (Chr(10)), or carriage return–linefeed character combination (Chr(13) & Chr(10)) between each line.
title	Optional. String expression displayed in the title bar of the dialog box. If you omit title , the application name is placed in the title bar.
default	Optional. String expression displayed in the text box as the default response if no other input is provided. If you omit default , the text box is displayed empty.
xpos	Optional. Numeric expression that specifies, in twips, the horizontal distance of the left edge of the dialog box from the left edge of the screen. If xpos is omitted, the dialog box is horizontally centered.
ypos	Optional. Numeric expression that specifies, in twips, the vertical distance of the upper edge of the dialog box from the top of the screen. If ypos is omitted, the dialog box is vertically positioned approximately one-third of the way down the screen.
helpfile	Optional. String expression that identifies the Help file to use to provide context-sensitive Help for the dialog box. If helpfile is provided, context must also be provided.

context Optional. Numeric expression that is the Help context number assigned to the appropriate Help topic by the Help author. If ***context*** is provided, ***helpfile*** must also be provided.

Remarks

When both ***helpfile*** and ***context*** are provided, the user can press F1 (Windows) or HELP (Macintosh) to view the Help topic corresponding to the ***context***. Some host applications, for example, Microsoft Excel, also automatically add a **Help** button to the dialog box. If the user clicks **OK** or presses ENTER, the **InputBox** function returns whatever is in the text box. If the user clicks **Cancel**, the function returns a zero-length string ("").

Note To specify more than the first named argument, you must use **InputBox** in an expression. To omit some positional arguments, you must include the corresponding comma delimiter.

MsgBox Function

Displays a message in a dialog box, waits for the user to click a button, and returns an **Integer** indicating which button the user clicked.

Syntax

MsgBox(*prompt* [, *buttons*] [, *title*] [, *helpfile*, *context*])

The **MsgBox** function syntax has these named arguments:

Part	Description
prompt	Required. String expression displayed as the message in the dialog box. The maximum length of prompt is approximately 1024 characters, depending on the width of the characters used. If prompt consists of more than one line, you can separate the lines using a carriage return character (Chr(13)), a linefeed character (Chr(10)), or carriage return – linefeed character combination (Chr(13) & Chr(10)) between each line.
buttons	Optional. Numeric expression that is the sum of values specifying the number and type of buttons to display, the icon style to use, the identity of the default button, and the modality of the message box. If omitted, the default value for buttons is 0.
title	Optional. String expression displayed in the title bar of the dialog box. If you omit title , the application name is placed in the title bar.
helpfile	Optional. String expression that identifies the Help file to use to provide context-sensitive Help for the dialog box. If helpfile is provided, context must also be provided.
context	Optional. Numeric expression that is the Help context number assigned to the appropriate Help topic by the Help author. If context is provided, helpfile must also be provided.

Settings

The **buttons** argument settings are:

Constant	Value	Description
vbOKOnly	0	Display OK button only.
vbOKCancel	1	Display OK and Cancel buttons.
vbAbortRetryIgnore	2	Display Abort , Retry , and Ignore buttons.
vbYesNoCancel	3	Display Yes , No , and Cancel buttons.
vbYesNo	4	Display Yes and No buttons.
vbRetryCancel	5	Display Retry and Cancel buttons.
vbCritical	16	Display Critical Message icon.
vbQuestion	32	Display Warning Query icon.
vbExclamation	48	Display Warning Message icon.
vbInformation	64	Display Information Message icon.
vbDefaultButton1	0	First button is default.
vbDefaultButton2	256	Second button is default.
vbDefaultButton3	512	Third button is default.
vbDefaultButton4	768	Fourth button is default.
vbApplicationModal	0	Application modal; the user must respond to the message box before continuing work in the current application.
vbSystemModal	4096	System modal; all applications are suspended until the user responds to the message box.
vbMsgBoxHelpButton	16384	Adds Help button to the message box
VbMsgBoxSetForeground	65536	Specifies the message box window as the foreground window
vbMsgBoxRight	524288	Text is right aligned
vbMsgBoxRtlReading	1048576	Specifies text should appear as right-to-left reading on Hebrew and Arabic systems

The first group of values (0–5) describes the number and type of buttons displayed in the dialog box; the second group (16, 32, 48, 64) describes the icon style; the third group (0, 256, 512) determines which button is the default; and the fourth group (0, 4096) determines the modality of the message box. When

adding numbers to create a final value for the *buttons* argument, use only one number from each group.

Note These constants are specified by Visual Basic for Applications. As a result, the names can be used anywhere in your code in place of the actual values.

Return Values

Constant Value Description

vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

Remarks

When both *helpfile* and *context* are provided, the user can press F1 (Windows) or HELP (Macintosh) to view the Help topic corresponding to the **context**. Some host applications, for example, Microsoft Excel, also automatically add a **Help** button to the dialog box.

If the dialog box displays a **Cancel** button, pressing the ESC key has the same effect as clicking **Cancel**. If the dialog box contains a **Help** button, context-sensitive Help is provided for the dialog box. However, no value is returned until one of the other buttons is clicked.

Note To specify more than the first named argument, you must use **MsgBox** in an expression. To omit some positional arguments, you must include the corresponding comma delimiter.

Choose Function

Selects and returns a value from a list of arguments.

Syntax

Choose(*index*, *choice-1*[, *choice-2*, ... [, *choice-n*]])

The **Choose** function syntax has these parts:

Part	Description
<i>index</i>	Required. Numeric expression or field that results in a value between 1 and the number of available choices.
<i>choice</i>	Required. Variant expression containing one of the possible choices.

Remarks

Choose returns a value from the list of choices based on the value of *index*. If *index* is 1, **Choose** returns the first choice in the list; if *index* is 2, it returns the second choice, and so on.

You can use **Choose** to look up a value in a list of possibilities. For example, if *index* evaluates to 3 and *choice-1* = "one", *choice-2* = "two", and *choice-3* = "three", **Choose** returns "three". This capability is particularly useful if *index* represents the value in an option group.

Choose evaluates every choice in the list, even though it returns only one. For this reason, you should watch for undesirable side effects. For example, if you use the **MsgBox** function as part of an expression in all the choices, a message box will be displayed for each choice as it is evaluated, even though **Choose** returns the value of only one of them.

The **Choose** function returns a Null if *index* is less than 1 or greater than the number of choices listed.

If *index* is not a whole number, it is rounded to the nearest whole number before

being evaluated.

IIf Function

Returns one of two parts, depending on the evaluation of an expression.

Syntax

IIf(*expr*, *truepart*, *falsepart*)

The **IIf** function syntax has these named arguments:

Part	Description
<i>expr</i>	Required. Expression you want to evaluate.
<i>truepart</i>	Required. Value or expression returned if <i>expr</i> is True .
<i>falsepart</i>	Required. Value or expression returned if <i>expr</i> is False .

Remarks

IIf always evaluates both *truepart* and *falsepart*, even though it returns only one of them. Because of this, you should watch for undesirable side effects. For example, if evaluating *falsepart* results in a division by zero error, an error occurs even if *expr* is **True**.

Switch Function

Evaluates a list of expressions and returns a **Variant** value or an expression associated with the first expression in the list that is **True**.

Syntax

Switch(*expr-1*, *value-1* [, *expr-2*, *value-2* ... [, *expr-n*, *value-n*]])

The **Switch** function syntax has these parts:

Part	Description
<i>expr</i>	Required. Variant expression you want to evaluate.
<i>value</i>	Required. Value or expression to be returned if the corresponding expression is True .

Remarks

The **Switch** function argument list consists of pairs of expressions and values. The expressions are evaluated from left to right, and the value associated with the first expression to evaluate to **True** is returned. If the parts aren't properly paired, a run-time error occurs. For example, if *expr-1* is **True**, **Switch** returns *value-1*. If *expr-1* is **False**, but *expr-2* is **True**, **Switch** returns *value-2*, and so on.

Switch returns a Null value if:

- None of the expressions is **True**.
- The first **True** expression has a corresponding value that is **Null**.

Switch evaluates all of the expressions, even though it returns only one of them. For this reason, you should watch for undesirable side effects. For example, if the evaluation of any expression results in a division by zero error, an error occurs.

Format Function

Returns a **Variant (String)** containing an expression formatted according to instructions contained in a format expression.

Syntax

Format(*expression*[, *format*[, *firstdayofweek*[, *firstweekofyear*]])

The **Format** function syntax has these parts:

Part	Description
<i>expression</i>	Required. Any valid expression.
<i>format</i>	Optional. A valid named or user-defined format expression.
<i>firstdayofweek</i>	Optional. A constant that specifies the first day of the week.
<i>firstweekofyear</i>	Optional. A constant that specifies the first week of the year.

Settings

The *firstdayofweek* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use NLS API setting.
VbSunday	1	Sunday (default)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

The *firstweekofyear* argument has these settings:

Constant	Value	Description
vbUseSystem	0	Use NLS API setting.
vbFirstJan1	1	Start with week in which January 1 occurs (default).
vbFirstFourDays	2	Start with the first week that has at least four days in the year.
vbFirstFullWeek	3	Start with the first full week of the year.

Remarks

To Format	Do This
Numbers	Use predefined named numeric formats or create user-defined numeric formats.
Dates and times	Use predefined named date/time formats or create user-defined date/time formats.
Date and time serial numbers	Use date and time formats or numeric formats.
Strings	Create your own user-defined string formats.

If you try to format a number without specifying *format*, **Format** provides functionality similar to the **Str** function, although it is internationally aware. However, positive numbers formatted as strings using **Format** don't include a leading space reserved for the sign of the value; those converted using **Str** retain the leading space.

If you are formatting a non-localized numeric string, you should use a user-defined numeric format to ensure that you get the look you want.

Note If the **Calendar** property setting is Gregorian and *format* specifies date formatting, the supplied *expression* must be Gregorian. If the Visual Basic **Calendar** property setting is Hijri, the supplied *expression* must be Hijri.

If the calendar is Gregorian, the meaning of *format* expression symbols is unchanged. If the calendar is Hijri, all date format symbols (for example, *dddd*, *mmmm*, *yyyy*) have the same meaning but apply to the Hijri calendar. Format symbols remain in English; symbols that result in text display (for example, AM and PM) display the string (English or Arabic) associated with that symbol. The

range of certain symbols changes when the calendar is Hijri.

Symbol	Range
<i>d</i>	1-30
<i>dd</i>	1-30
<i>ww</i>	1-51
<i>mmm</i>	Displays full month names (Hijri month names have no abbreviations).
<i>y</i>	1-355
<i>yyyy</i>	100-9666

InStr Function

Returns a **Variant (Long)** specifying the position of the first occurrence of one string within another.

Syntax

InStr([*start*,]*string1*, *string2*[, *compare*])

The **InStr** function syntax has these arguments:

Part	Description
<i>start</i>	Optional. Numeric expression that sets the starting position for each search. If omitted, search begins at the first character position. If <i>start</i> contains Null, an error occurs. The <i>start</i> argument is required if <i>compare</i> is specified.
<i>string1</i>	Required. String expression being searched.
<i>string2</i>	Required. String expression sought.
<i>compare</i>	Optional. Specifies the type of string comparison. If <i>compare</i> is Null, an error occurs. If <i>compare</i> is omitted, the Option Compare setting determines the type of comparison. Specify a valid LCID (LocaleID) to use locale-specific rules in the comparison.

Settings

The *compare* argument settings are:

Constant	Value	Description
vbUseCompareOption	-1	Performs a comparison using the setting of the Option Compare statement.
vbBinaryCompare	0	Performs a binary comparison.
vbTextCompare	1	Performs a textual comparison.
vbDatabaseCompare	2	Microsoft Access only. Performs a comparison based on information in your database.

Return Values

If	InStr returns
<i>string1</i> is zero-length	0
<i>string1</i> is Null	Null
<i>string2</i> is zero-length	start
<i>string2</i> is Null	Null
<i>string2</i> is not found	0
<i>string2</i> is found within <i>string1</i>	Position at which match is found
<i>start</i> > <i>string2</i>	0

Remarks

The **InStrB** function is used with byte data contained in a string. Instead of returning the character position of the first occurrence of one string within another, **InStrB** returns the byte position.

InStrRev Function

Description

Returns the position of an occurrence of one string within another, from the end of string.

Syntax

InStrRev(*stringcheck*, *stringmatch*[, *start*[, *compare*]])

The **InStrRev** function syntax has these named arguments:

Part	Description
<i>stringcheck</i>	Required. String expression being searched.
<i>stringmatch</i>	Required. String expression being searched for.
<i>start</i>	Optional. Numeric expression that sets the starting position for each search. If omitted, -1 is used, which means that the search begins at the last character position. If <i>start</i> contains Null, an error occurs.
<i>compare</i>	Optional. Numeric value indicating the kind of comparison to use when evaluating substrings. If omitted, a binary comparison is performed. See Settings section for values.

Settings

The *compare* argument can have the following values:

Constant	Value	Description
vbUseCompareOption	-1	Performs a comparison using the setting of the Option Compare statement.
vbBinaryCompare	0	Performs a binary comparison.
vbTextCompare	1	Performs a textual comparison.
		Microsoft Access only. Performs a comparison

vbDatabaseCompare 2 based on information in your database.

Return Values

InStrRev returns the following values:

If	InStrRev returns
<i>stringcheck</i> is zero-length	0
<i>stringcheck</i> is Null	Null
<i>stringmatch</i> is zero-length	<i>start</i>
<i>stringmatch</i> is Null	Null
<i>stringmatch</i> is not found	0
<i>stringmatch</i> is found within <i>stringcheck</i>	Position at which match is found
<i>start</i> > Len(stringmatch)	0

Remarks

Note that the syntax for the **InstrRev** function is not the same as the syntax for the **Instr** function.

LCASE Function

Returns a String that has been converted to lowercase.

Syntax

LCASE(*string*)

The required *string* argument is any valid string expression. If *string* contains Null, Null is returned.

Remarks

Only uppercase letters are converted to lowercase; all lowercase letters and nonletter characters remain unchanged.

Left Function

Returns a **Variant (String)** containing a specified number of characters from the left side of a string.

Syntax

Left(*string*, *length*)

The **Left** function syntax has these named arguments:

Part	Description
<i>string</i>	Required. String expression from which the leftmost characters are returned. If <i>string</i> contains Null, Null is returned.
<i>length</i>	Required; Variant (Long) . Numeric expression indicating how many characters to return. If 0, a zero-length string ("") is returned. If greater than or equal to the number of characters in <i>string</i> , the entire string is returned.

Remarks

To determine the number of characters in *string*, use the **Len** function.

Note Use the **LeftB** function with byte data contained in a string. Instead of specifying the number of characters to return, *length* specifies the number of bytes.

Len Function

Returns a Long containing the number of characters in a string or the number of bytes required to store a variable.

Syntax

Len(*string* | *varname*)

The **Len** function syntax has these parts:

Part	Description
<i>string</i>	Any valid string expression. If <i>string</i> contains Null, Null is returned.
	Any valid variable name. If <i>varname</i> contains Null , Null is returned.
<i>Varname</i>	If <i>varname</i> is a Variant, Len treats it the same as a String and always returns the number of characters it contains.

Remarks

One (and only one) of the two possible arguments must be specified. With user-defined types, **Len** returns the size as it will be written to the file.

Note Use the **LenB** function with byte data contained in a string, as in double-byte character set (DBCS) languages. Instead of returning the number of characters in a string, **LenB** returns the number of bytes used to represent that string. With user-defined types, **LenB** returns the in-memory size, including any padding between elements. For sample code that uses **LenB**, see the second example in the example topic.

Note **Len** may not be able to determine the actual number of storage bytes required when used with variable-length strings in user-defined data types.

LTrim, RTrim, and Trim Functions

Returns a **Variant (String)** containing a copy of a specified string without leading spaces (**LTrim**), trailing spaces (**RTrim**), or both leading and trailing spaces (**Trim**).

Syntax

LTrim(*string*)

RTrim(*string*)

Trim(*string*)

The required *string* argument is any valid string expression. If *string* contains Null, **Null** is returned.

Mid Function

Returns a **Variant (String)** containing a specified number of characters from a string.

Syntax

Mid(*string*, *start*[, *length*])

The **Mid** function syntax has these named arguments:

Part	Description
<i>string</i>	Required. String expression from which characters are returned. If <i>string</i> contains Null, Null is returned.
<i>start</i>	Required; Long. Character position in <i>string</i> at which the part to be taken begins. If <i>start</i> is greater than the number of characters in <i>string</i> , Mid returns a zero-length string ("").
<i>length</i>	Optional; Variant (Long) . Number of characters to return. If omitted or if there are fewer than <i>length</i> characters in the text (including the character at <i>start</i>), all characters from the <i>start</i> position to the end of the string are returned.

Remarks

To determine the number of characters in *string*, use the **Len** function.

Note Use the **MidB** function with byte data contained in a string, as in double-byte character set languages. Instead of specifying the number of characters, the arguments specify numbers of bytes. For sample code that uses **MidB**, see the second example in the example topic.

Replace Function

Description

Returns a string in which a specified substring has been replaced with another substring a specified number of times.

Syntax

Replace(*expression*, *find*, *replace*[, *start*[, *count*[, *compare*]]])

The **Replace** function syntax has these named arguments:

Part	Description
<i>expression</i>	Required. String expression containing substring to replace.
<i>find</i>	Required. Substring being searched for.
<i>replace</i>	Required. Replacement substring.
<i>start</i>	Optional. Position within <i>expression</i> where substring search is to begin. If omitted, 1 is assumed.
<i>count</i>	Optional. Number of substring substitutions to perform. If omitted, the default value is -1, which means make all possible substitutions.
<i>compare</i>	Optional. Numeric value indicating the kind of comparison to use when evaluating substrings. See Settings section for values.

Settings

The *compare* argument can have the following values:

Constant	Value	Description
vbUseCompareOption	-1	Performs a comparison using the setting of the Option Compare statement.
vbBinaryCompare	0	Performs a binary comparison.
vbTextCompare	1	Performs a textual comparison.
		Microsoft Access only. Performs a comparison

vbDatabaseCompare 2 based on information in your database.

Return Values

Replace returns the following values:

If	Replace returns
<i>expression</i> is zero-length	Zero-length string ("")
<i>expression</i> is Null	An error.
<i>find</i> is zero-length	Copy of <i>expression</i> .
<i>replace</i> is zero-length	Copy of <i>expression</i> with all occurrences of <i>find</i> removed.
<i>start</i> > Len(expression)	Zero-length string.
<i>count</i> is 0	Copy of <i>expression</i> .

Remarks

The return value of the **Replace** function is a string, with substitutions made, that begins at the position specified by *start* and concludes at the end of the *expression* string. It is not a copy of the original string from start to finish.

Right Function

Returns a **Variant (String)** containing a specified number of characters from the right side of a string.

Syntax

Right(*string*, *length*)

The **Right** function syntax has these named arguments:

Part	Description
<i>string</i>	Required. String expression from which the rightmost characters are returned. If <i>string</i> contains Null, Null is returned.
<i>length</i>	Required; Variant (Long) . Numeric expression indicating how many characters to return. If 0, a zero-length string ("") is returned. If greater than or equal to the number of characters in <i>string</i> , the entire string is returned.

Remarks

To determine the number of characters in *string*, use the **Len** function.

Note Use the **RightB** function with byte data contained in a string. Instead of specifying the number of characters to return, *length* specifies the number of bytes.

Space Function

Returns a **Variant (String)** consisting of the specified number of spaces.

Syntax

Space(*number*)

The required *number* argument is the number of spaces you want in the string.

Remarks

The **Space** function is useful for formatting output and clearing data in fixed-length strings.

StrComp Function

Returns a **Variant (Integer)** indicating the result of a string comparison.

Syntax

StrComp(*string1*, *string2*[, *compare*])

The **StrComp** function syntax has these named arguments:

Part	Description
<i>string1</i>	Required. Any valid string expression.
<i>string2</i>	Required. Any valid string expression.
<i>compare</i>	Optional. Specifies the type of string comparison. If the <i>compare</i> argument is Null, an error occurs. If <i>compare</i> is omitted, the Option Compare setting determines the type of comparison.

Settings

The **compare** argument settings are:

Constant	Value	Description
vbUseCompareOption	-1	Performs a comparison using the setting of the Option Compare statement.
vbBinaryCompare	0	Performs a binary comparison.
vbTextCompare	1	Performs a textual comparison.
vbDatabaseCompare	2	Microsoft Access only. Performs a comparison based on information in your database.

Return Values

The **StrComp** function has the following return values:

If	StrComp returns
----	-----------------

string1 is less than ***string2*** -1
string1 is equal to ***string2*** 0
string1 is greater than ***string2*** 1
string1 or ***string2*** is Null Null

String Function

Returns a **Variant (String)** containing a repeating character string of the length specified.

Syntax

String(*number*, *character*)

The **String** function syntax has these named arguments:

Part	Description
<i>number</i>	Required; Long. Length of the returned string. If <i>number</i> contains Null, Null is returned.
<i>character</i>	Required; Variant. Character code specifying the character or string expression whose first character is used to build the return string. If <i>character</i> contains Null , Null is returned.

Remarks

If you specify a number for *character* greater than 255, **String** converts the number to a valid character code using the formula:

character Mod 256

StrReverse Function

Description

Returns a string in which the character order of a specified string is reversed.

Syntax

StrReverse(*expression*)

The *expression* argument is the string whose characters are to be reversed. If *expression* is a zero-length string (""), a zero-length string is returned. If *expression* is **Null**, an error occurs.

UCase Function

Returns a **Variant (String)** containing the specified string, converted to uppercase.

Syntax

UCase(*string*)

The required *string* argument is any valid string expression. If *string* contains Null, **Null** is returned.

Remarks

Only lowercase letters are converted to uppercase; all uppercase letters and nonletter characters remain unchanged.

[Show All](#)

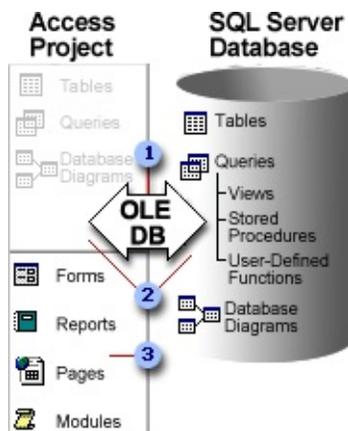
About an Access project (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Some of the content in this topic may not be applicable to some languages.

[What is a Microsoft Access Project?](#)

A Microsoft Access project (.adp) is an Access data file that provides efficient, native-mode access to a Microsoft SQL Server database through the [OLE DB](#) component architecture. Using an Access project, you can create a client/server application as easily as a file server application. This client/server application can be a traditional solution based on forms and reports, or a Web-based solution based on [data access pages](#), or a combination of both. You can connect the Access project to a remote SQL Server database, a local SQL Server database, or a local installation of SQL Server 2000 Desktop engine.



- 1 Connects the Access project to the SQL Server database
- 2 Database objects stored in the SQL Server database
- 3 Database objects stored in the Access project

An Access project is called a project because it contains only code-based or HTML-based database objects: forms, reports, the name and location of data

access pages, macros, and modules. These are the database objects that you use to create an application. Unlike a [Microsoft Access database](#), an Access project does not contain any data or data definition based objects: [tables](#), [views](#), [database diagrams](#), [stored procedures](#), or [user-defined functions](#). Instead, these database objects are stored in the SQL Server database.

To create your application and access the data, you connect the Access project to the SQL Server database by using the **New** command on the **File** menu to display the **Data Link Properties** dialog box or by using the Database Wizard.

[Supported Microsoft SQL Server databases](#)

You can connect to one of the following SQL Server databases:

- Microsoft SQL Server 2000 on Microsoft Windows 2000 or later
- Microsoft SQL Server 2000 Desktop Engine on Windows 2000 or later

Note If you are connecting to a named instance of a SQL Server 2000 server, your Access project client computer must have Microsoft Data Access Components (MDAC) version 2.6 installed. You can install MDAC 2.6 by installing Microsoft SQL Server 2000 Personal Desktop Edition from the \MSDE2000 folder on the Microsoft Office 2003 CD-ROM.

[Working with an Access project](#)

Working with an Access project is very similar to working with an Access database. The process of creating forms, reports, data access pages, macros, and modules is virtually the same as that used to create an Access database.

Once you connect to an SQL Server database, you can view, create, modify, and delete tables, views, stored procedures, user-defined functions, and database diagrams by using the Database Designer, Table Designer, Query Designer, Query Builder, and SQL Text Editor. Although the user interface for working with these database objects is different from the equivalent database objects in an Access database, it is just as user-friendly. An Access project also contains many of the same wizards as an Access database, such as the Form Wizard, the Report Wizard, the Page Wizard, and the Input Mask Wizard. These Wizards help you to quickly create a prototype or simple application, and also make it

easier to create an advanced application.

[About extended properties](#)

If your Access project is connected to an SQL Server 2000 database, you can use and take advantage of extended properties. Extended properties store additional attribute information about SQL Server database objects that are stored in the SQL Server database and provide added functionality for an Access project such as:

- Creating a lookup on a column.
- Displaying user-friendly validation text.
- Formatting data and Defining input masks.
- Using subdatasheets, saving sort order and filters, and saving datasheet appearance of tables, views, stored procedures, and inline user-defined functions.

Some extended properties are used by Access and are not visible through an Access project, such as a property which sets the column widths in a table datasheet. Other extended properties are available from the Database, Table, and Query Designers property sheets for views, single statement stored procedures (except column properties), and in-line functions. Because multistatement stored procedures, and table and scalar user-defined functions are edited in the SQL Text Editor, they don't support extended properties.

By default, SQL Server does not support extended property inheritance. A column in a view does not automatically have the same properties as the column in the base table. However, an Access project does support inheritance by copying the extended properties from table columns to view and function columns on creation, and by copying extended properties from table columns to form and report control properties.

Note Changes to the extended properties of tables, views, stored procedures, and in-line functions are seen by all users of the database, because extended properties are stored in the SQL Server database. This means, for example, that if you change the format of a table datasheet, you are changing it for everyone who subsequently opens the table datasheet.

[Show All](#)

About differences between data types in an Access database and Access project

[About Access database data types](#)

Decide what kind of [data type](#) to use for a field based on these considerations:

- What kind of values do you want to allow in the field? For example, you can't store text in a field with a Number data type.
- How much storage space do you want to use for values in the field?
- What types of operations do you want to perform on the values in the field? For example, Microsoft Access can sum values in Number or Currency fields, but not values in [Text](#) or [OLE object](#) fields.
- Do you want to sort or [index](#) a field? OLE Object fields can't be sorted or indexed.
- Do you want to use a field to group records in queries or reports? OLE Object fields can't be used to group records.
- How do you want to sort values in a field? In a Text field, numbers sort as strings of characters (1, 10, 100, 2, 20, 200, and so on), not as numeric values. Use a Number or Currency field to sort numbers as numeric values. Also, many date formats will not sort properly if entered in a Text field. Use a Date/Time field to ensure proper sorting for dates.
- Will you need to store Microsoft Word or Microsoft Excel documents, pictures, sound, and other types of binary data created in other programs? OLE objects can be [linked](#) to or [embedded](#) in an OLE Object field in a Microsoft Access table. To display the OLE object, use a [control](#) in a form or report.

[Choosing between a Text or Memo field](#)

Microsoft Access provides two [field data types](#) to store data with text or combinations of text and numbers: Text and Memo.

Use a Text data type to store data such as names, addresses, and any numbers that do not require calculations, such as phone numbers, part numbers, or postal codes. A Text field can store up to 255 characters, but the default field size is 50 characters. The **FieldSize** property controls the maximum number of characters that can be entered in a Text field.

Use the Memo data type if you need to store more than 255 characters. A Memo field can store up to 65,536 characters. If you want to store formatted text or long documents, you should create an OLE Object field instead of a Memo field.

Both Text and Memo data types store only the characters entered in a field; space characters for unused positions in the field aren't stored.

You can sort or group on a Text field or a Memo field, but Access only uses the first 255 characters when you sort or group on a Memo field.



[Choosing between a Number or Currency field](#)

Microsoft Access provides two [field data types](#) to store data containing numeric values: Number and Currency.

Use a Number field to store numeric data to be used for mathematical calculations, except calculations that involve money or that require a high degree of accuracy. The kind and size of numeric values that can be stored in a Number field is controlled by setting the **FieldSize** property. For example, the Byte field size will only store whole numbers (no decimal values) from 0 to 255 and occupies 1 byte of disk space.

Use a Currency field to prevent rounding off during calculations. A Currency field is accurate to 15 digits to the left of the decimal point and 4 digits to the right. A Currency field occupies 8 bytes of disk space.

Number and Currency fields provide predefined [display formats](#), or you can create a custom format.



[Choosing between an incremented or random AutoNumber field](#)

Microsoft Access provides the AutoNumber data type to create fields that automatically enter a unique number when a record is added. Once a number is

generated for a record, it can't be deleted or changed. An AutoNumber field can generate three kinds of numbers: sequential numbers that increment by one, random numbers, and Replication ID (also referred to as GUIDs— globally unique identifiers) numbers. AutoNumbers that increment by one are the most common kind of AutoNumber and are a good choice for use as a table's [primary key](#). Random AutoNumbers will generate a random number that is unique to each record within the table. Replication ID AutoNumbers are used in [database replication](#) to generate unique identifiers for synchronizing [replicas](#).



[About Access project \(SQL Server\) data types](#)

Objects that contain data have an associated data type that defines the kind of data (character, integer, binary, and so on) the object can contain. The following objects have data types:

- Columns in tables and views.
- Parameters in stored procedures.
- Variables.
- Transact-SQL functions that return one or more data values of a specific data type.
- Stored procedures that have a return code, which always has an integer data type.

Assigning a data type to an object defines four attributes of the object:

- The kind of data contained by the object. For example, character, integer or binary.
- The length of the stored value, or its size. The length of an image, binary, and varbinary data type is defined in bytes. The length of any of the numeric data types is the number of bytes required to hold the number of digits allowed for that data type. The length of the character string and Unicode data types is defined in characters.
- The precision of the number (numeric data types only). The precision is the number of digits the number can contain. For example, a smallint object can hold a maximum of 5 digits; it has a precision of 5.
- The scale of the number (numeric data types only). The scale is the number of digits that can be stored to the right of the decimal point. For example, an int object cannot accept a decimal point and has a scale of 0. A money object can have a maximum of 4 digits to the right of the decimal point and

has a scale of 4.

For example, if an object is defined as money, it can contain a maximum of 19 digits, 4 of which can be to the right of the decimal. The object uses 8 bytes to store the data. The money data type therefore has a precision of 19, a scale of 4, and a length of 8.

All data stored in SQL Server must be compatible with one of these base data types. The cursor data type is the only base data type that cannot be assigned to a table column. It can be used only for variables and stored procedure parameters.

User-defined data types can also be created, for example:

```
-- Create a birthday datatype that allows nulls.  
EXEC sp_addtype birthday, datetime, 'NULL'
```

```
-- Create a table using the new data type.  
CREATE TABLE employee  
emp_id char(5)  
emp_first_name char(30)  
emp_last_name char(40)  
emp_birthday birthday
```

User-defined data types are always defined in terms of a base data type. They provide a mechanism for applying a name to a data type that is more descriptive of the types of values to be held in the object. This can make it easier for a programmer or database administrator to understand the intended use of any object defined with the data type.

 [Comparison or mapping of data types between an Access database and Access project](#)

The following table compares [data types](#) between a [Microsoft Access database](#) and a [Microsoft Access project](#).

Microsoft Access data type	SQL Server data type
Yes/No	bit
Number (Byte)	tinyint

Number (Integer)	smallint
Number (Long Integer)	int
Number (Single)	real
(no equivalent)	bigint
Number (Double)	float
	money
Currency	smallmoney
	decimal
Decimal/numeric	numeric
	datetime
Date/Time	smalldatetime
AutoNumber (Increment)	int (with the Identity property defined)
	varchar(n)
Text (n)	nvarchar(n)
Memo	text
OLE Object	image
Replication ID (also called globally unique identifier (GUID))	uniqueidentifier (SQL Server 7.0 or later)
Hyperlink	char , nchar , varchar , nvarchar (With the Hyperlink property set to Yes)
(no equivalent)	varbinary
(no equivalent)	smallint
(no equivalent)	timestamp
	char
(no equivalent)	nchar
(no equivalent)	sql_variant
(no equivalent)	user-defined

Note In an Access project or SQL Server database, the "n" prefix stands for "national" and means that the data type is unicode-enabled. In an Access database, all text columns are unicode-enabled by default.

[Show All](#)

About SQL Server 2000 Desktop Engine

Some of the content in this topic may not be applicable to some languages.

Note The previous version of SQL Server 2000 Desktop Engine was called Microsoft Data Engine (MSDE).



[What is SQL Server 2000 Desktop Engine?](#)

Microsoft SQL Server 2000 Desktop Engine is a technology that provides local data storage in a format compatible with Microsoft SQL Server 2000. You can also use SQL Server 2000 Desktop Engine as a remote data storage solution. Think of SQL Server 2000 Desktop Engine as a client/server data engine alternative to the file server [Microsoft Jet database engine](#). SQL Server 2000 Desktop Engine runs under Microsoft Windows 2000 or later. It is designed and optimized for use on smaller computer systems, such as a single-user computer or small workgroup server.

Because SQL Server 2000 Desktop Engine is based on the same data engine as SQL Server, most [Microsoft Access projects](#) or client/server applications run on either version. However unlike SQL Server, SQL Server 2000 Desktop Engine has a 2 gigabyte database size limit, and when using transactional replication, can't be a replication publisher (although it can act as a replication subscriber).

Note Do not confuse SQL Server 2000 Desktop Engine with SQL Server 2000 Personal Edition, which includes a full set of management tools and most of the functionality of SQL Server Standard Edition, but is optimized for personal use and is a separate product.



[When to use SQL Server 2000 Desktop Engine](#)

Consider using SQL Server 2000 Desktop Engine as a desktop database alternative to a [Microsoft Access database](#) in the following ways:

- As a small workgroup server database. You can develop your applications by using an Access project connected to SQL Server 2000 Desktop Engine, if you anticipate that your workgroup and its business requirements may grow over time until they eventually need the full functionality of SQL Server 2000 running on a larger network server. In general, SQL Server 2000 Desktop Engine can handle a user workload similar to the [Microsoft Jet database engine](#) of an Access database. However, SQL Server 2000 Desktop Engine has a limit of five simultaneous active query batches (also called threads). Subsequent query batches are queued by SQL Server 2000 Desktop Engine and executed when the next batch is available. If you want to see how many times you have reached this limit in a SQL Server 2000 database, you can execute the DBCC CONCURRENTYVIOLATION command. For more information on workloads, batches, and DBCC, see the SQL Server documentation.
- As a replication subscriber in an SQL Server replicated database environment. For example, you may have mobile users working remotely from the main office who access and update SQL Server 2000 Desktop Engine on laptops or disconnected computers, but then periodically connect to the Master SQL Server database to reconcile changes.
- To easily develop and test an Access project or client/server application on a personal computer or workstation, and then modify the Access project connection information to connect to an SQL Server database on a remote server for final testing and production.
- As the local subscription database for offline data.

 [About using SQL Server 2000 Desktop Engine database utilities from Access](#)

Microsoft SQL Server 2000 Desktop Engine offers the basic database engine features of SQL Server 2000 but does not include a user interface, management tools, analysis capabilities, merge replication support, client access licenses, developer libraries, or SQL Server Books Online. It also limits database size and user workload. It has the smallest footprint of any edition of SQL Server 2000 and is thus an ideal embedded or offline data store.

From a [Microsoft Access project](#), you can perform some common administrative tasks on a Microsoft SQL Server database.

Command

Description

Back Up SQL Database	Creates a backup file (.dat) of the current SQL Server database.
Restore SQL Database	Restores a SQL Server database from a backup database file (.dat).
Drop SQL Database	Deletes a SQL Server database.
Copy Database File	Copies a SQL Server database 7.0 or later database to another SQL Server.
Transfer Database	Transfers a SQL Server Database version 7.0 or later database file (.mdf).

These commands are visible from the **Database Utilities** command on the **Tools** menu when you install Microsoft Office 2003, but only if you have installed SQL Server 2000 Desktop Engine on your computer.

[Show All](#)

Install and configure SQL Server 2000 Desktop Engine

Although Microsoft SQL Server 2000 Desktop Engine is not part of a standard Microsoft Office 2003 installation, you can install SQL Server 2000 Desktop Engine from the Office 2003 CD-ROM. This applies to computers running Windows 2000 or later.

[Install SQL Server 2000 Desktop Engine](#)

1. If you have Microsoft Data Engine (MSDE) installed, uninstall it first before installing the SQL Server 2000 Desktop Engine.
2. Run the MSDE2KS3.EXE self-extracting program in the \MSDE2000 folder on the Office 2003 CD-ROM, or on the network location at your local site if Office 2003 was installed as a network installation.

By default, the files will be extracted to the C:\sql2ksp3 folder. You can specify another location, but the following instructions are based on this default location.

3. After the self-extraction is complete, go to the C:\sql2ksp3\msde folder.
4. Do one of the following:
 - If you are installing SQL Server 2000 Desktop Engine for the first time, you need to use a command-line switch with setup.exe. Click **Run** on the **Start** menu, type **CMD** and press ENTER to bring up the DOS command window. Type the following commands:

```
CD \sql2ksp3\msde
setup.exe SAPWD="AStrongSAPwd"
```

- If you are upgrading an existing instance of SQL Server 2000 Desktop Engine that was included with Microsoft Office XP or Microsoft Access 2002, you need to use command-line switches with setup.exe. Click **Run** on the **Start** menu, type **CMD** and press ENTER to bring up the DOS command window. Type the following commands:

```
CD \sql2ksp3\msde
setup.exe /upgradesp setup\sqlrun01.msi SAPWD="AStrongSAPwd"
```

Replace "**AStrongSAPwd**" with your own strong password for the system administrator (SA) login account.

5. Restart the operating system to complete the installation.
6. When the installation or upgrade is complete, you can delete the files in the C:\sql2ksp3 folder.

Note For additional information about installing SQL Server 2000 Desktop Engine, refer to the SP3ReadMe.htm in the \sql2ksp3\msde folder.

 [About enabling the system administrator \(SA\) user name in an Access project](#)

When the Microsoft SQL Sever 2000 Desktop Engine (formerly called the Microsoft Database Engine or MSDE) is installed on Microsoft Windows computers, it is installed with Windows NT Authentication implemented (this feature is also known as integrated security). When a user connects to the SQL Server database through a Microsoft Access project, the connection is enabled through a Windows user account. SQL Server verifies that the account name and password were validated when the user logged on to the system and grants access to the database, without requiring a separate logon name or password.

With the default installation of the Microsoft SQL Server 2000 Desktop Engine with integrated security, the user must be a member of the administrator group for the machine on which the Access project resides. This makes it difficult to allow other users to gain access to the Access project.

There is currently no simple way for the system administrator to create new logon accounts to the locally installed SQL Server database except by using SQL Tools or Transact-SQL (TSQL) commands. See the Microsoft Developer Network Web site for more information on TSQL.

You can change the security mode for accessing the SQL Server database on a computer running Windows. By setting properties from the **View** menu, you can enable mixed mode security. Mixed mode security allows access to the database

from either a Windows account or through a SQL Server account by using the default system administrator (SA) account with no password.

Using the default SA account, users can access the SQL Server database through the Access project without any additional security requirements. The administrator of the SQL Server can also add additional security by changing the default SA account password.

In order for the **Enable system administrator (SA) user name** check box to be enabled, the following must be true.

- The Access project must be connected to a SQL Server 7.0 (or later) database or a Microsoft SQL Server 2000 Desktop Engine.
- The SQL Server database must be running on a Windows platform.
- The SQL Server must be running on the same computer as the Access project.
- The current Windows user must be an Administrator for the computer.
- The current SQL Server user must be a member of the System Administrator role.

Note After changing the security mode, it is strongly recommended that the SA password be changed by using the **Set Login Password** command (on the **Tools** menu, point to **Security**).

[Start the SQL Server Service Manager](#)

Once you install Microsoft SQL Server 2000 Desktop Engine on your computer, you may need to start the SQL Server Service Manager. SQL Server 2000 Desktop Engine starts automatically on Windows 2000 or later.

1. Double-click the **MSSQLServer** icon in the icon tray of your windows task bar.

The SQL Server Service Manager window is displayed.

2. Click **Start/Continue**.
3. If you want SQL Server Service Manager to automatically start each time you start Windows, select the **Auto-start service when OS starts** check box.

[Customizing the installation](#)

You can do a custom installation of SQL Server 2000 Desktop Engine by using the following configuration options either in the Microsoft Windows setup.ini file, or as command line switches when you run setup.exe.

Note All configuration options must be in upper-case.

Setting	Description
TARGETDIR	The name of the folder to install SQL Server 2000 Desktop Engine. The default value is C:\Program Files\Microsoft SQL Server\.
DATADIR	The name of an optional folder for data files. The default value is the value of TARGETDIR.
COLLATION	The name of a collation sequence.
INSTANCENAME	The name of the SQL Server. The Default value is MSSQLSERVER.
SAPWD	A new password to replace the default password for the system administrator (SA) login account.

The following is an example of a custom installation using command line switches.

```
Setup.exe \\MySwdisk\SQLSW\ TARGETDIR="C:\Program Files\SQL8\  
DATADIR="C:\Program Files\SQLDATA\  
COLLATION="SQL_Latin1_General_CP1257_CS_AS" INSTANCENAME=myinstance
```

[Show All](#)

Ways to deploy an Access project

There are several common ways to deploy a [Microsoft Access project](#) (.adp) and the Microsoft SQL Server database connected to it.

[Using Access projects in a multiuser environment](#)

In a multiuser environment, each user often needs his or her own copy of the Access project file that is connected to the same Microsoft SQL Server database. You cannot modify objects, such as forms and reports, in the same Access project file among two or more users.

When you open an Access project, it's always opened in exclusive mode. If another user has opened an Access project first, and you try to open the same Access project while that user still has it open, Access alerts you that the file is opened exclusively and gives you the option of opening a read-only copy. In this case, objects in the Access project can't be modified.

In practice, when you are running a shared application you often provide a copy of the Access project file to each user of the application. In this case, each user has his or her own copy of forms, reports, data access pages, macros, and modules.

Whether or not a user has a personal copy of an Access project, one SQL Server database and the tables, queries, and database diagrams that reside in that database are in fact being shared. Therefore, it's important in a multiuser environment to also provide adequate security on these shared database objects so their integrity is not compromised.

If your users are using Microsoft Windows Terminal Server, each user can have a copy of the Access project file in his or her individual user data space defined in the user's profile on the Terminal Server. You can copy the Access project into each user's default application data folder and when a user logs in through Terminal Server, each user gets a personal copy of the Access project file.

[Sharing a Microsoft SQL database on the internet](#)

You can create data access pages or output one or more database objects to server-generated HTML or static HTML. You can then display these Web pages in Microsoft Internet Explorer, and access data through these Web pages.

[Creating an offline data application](#)

You don't always have online access to your company databases. However, you often want to take selective data offline (that is, download the data to a computer, such as a laptop or home computer), review and update the data away from your workplace, and then conveniently update and reconcile data changes when you have online access again. Using a combination of Microsoft products, a developer and database administrator can create an application that enables offline data access for users.

[Distributing your Access project file and database to users](#)

Many application projects with a single database can make use of several copies of the database: a development database, a test database, and a deployed production database. During the lifetime of a deployed database application, changes and improvements are first made in the development database, then propagated to the test database, and then propagated to the production database.

You might want to create an Access project and, after it is ready for testing or production, disconnect it from the SQL Server database. For example, you can create an Access project application and wait until after you have distributed copies of it to connect it to a Microsoft SQL Server database. You can distribute your Access project file to users, and then give them instructions on how to first open the file and then use the **Connection** command on the **File** menu to connect to the server where your database is located.

You can also use the **Copy SQL Database** or **Transfer SQL database** commands, macro actions, or methods to make the databases available to users.

Alternatively you could build the SQL Server database with a script file. The Northwind Traders sample Access project (NorthwindCS.adp) uses Visual Basic for Applications (VBA) code in the Startup module (which is invoked by using a function call from the OnOpen event of the Startup form) to build the SQL Server 2000 database by using Transact-SQL code contained in a script file (NorthwindCS.SQL), and then attaches the database to SQL Server. The VBA

code in the Startup module uses a combination of Access, ADO, and SQL-DMO objects to perform these operations.

[Using the Package and Deployment Wizard](#)

If you have Microsoft Office Access 2003 Developer Extensions, you can use the Package and Deployment Wizard to create an installation program that will install both an Access project application and a SQL Server 2000 Desktop Engine database. For information about deploying Access client/server applications with this wizard, see the Access 2003 Developer Extensions Help files.

[Replicating an SQL Server database](#)

If you use two computers, such as an office computer and a portable computer, you can make replicas of your SQL Server database and keep those replicas synchronized. Many users at different locations can work on personal database copies of a Master database at the same time and then synchronize the copies over the network, either through a dial-up connection or the Internet. For more information about SQL Server replication, see the SQL Server documentation.

[Show All](#)

Comparison of functions between Visual Basic for Applications and SQL Server Transact-SQL (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The following tables compare commonly-used functions in Microsoft Visual Basic for Applications (VBA) and Microsoft SQL Server Transact-SQL. For more information on Transact-SQL scalar functions, see the SQL Server documentation.

Note The Upsizing Wizard converts the following VBA functions (which can occur in table validation rules, field validation, and defaults) to their equivalent Transact-SQL functions.

[String functions](#)

VBA functions	Transact-SQL scalar functions
Asc(x)	ASCII(x)
Chr\$(x)	CHAR(x) LCASE(x)
Lcase\$(x)	LOWER(x)
Len(x)	DATALENGTH(x)
Ltrim\$(x)	LTRIM(x)
Mid\$(x,y,z)	SUBSTRING(x,y,z)
Right\$(x,y)	RIGHT(x,y)
Rtrim\$(x)	RTRIM(x)
Space\$(x)	SPACE(x)
Str\$(x)	STR(x)

	UCASE(x)
Ucase\$(x)	UPPER(x)

[Conversion functions](#)

VBA functions	Transact-SQL scalar functions
Ccur(x)	CONVERT(money,x)
Cdbl(x)	CONVERT(float,x)
Cint(x)	CONVERT(smallint,x)
Clng(x)	CONVERT(int,x)
Csng(x)	CONVERT(real,x)
Cstr(x)	CONVERT(varchar,x)
Cvdate(x)	CONVERT(datetime,x)

[Date functions](#)

VBA functions	Transact-SQL scalar functions
Date(x)	CONVERT(DATETIME,CONVERT(varchar,GETDATE(x)))
Dateadd(" <Access datepart>",x,y)	DATEADD(<SQL Server datepart>, x, y)
Datediff(" <Access datepart>",x,y)	DATEDIFF(<SQL Server datepart>, x, y)
Datepart(" <Access datepart>", x)	DATEPART(<SQL Server datepart>, x)
Day(x)	DATEPART(dd,x)
Hour(x)	DATEPART(hh,x)
Minute(x)	DATEPART(mi,x)
Month(x)	DATEPART(mm,x)

Now(x)	GETDATE(x)
Second(x)	DATEPART(ss,x)
Weekday(x)	DATEPART(dw,x)
Year(x)	DATEPART(yy,x)

[Math functions](#)

VBA functions	Transact-SQL scalar functions
----------------------	--

Int(x)	FLOOR(x)
Sgn(x)	SIGN(x)
Rnd(x)	ROUND(x)

[Show All](#)

About OLE DB and Data Links

Microsoft Access 2002 or later provides a native [OLE DB](#) user and programming interface that you can use to access a [Microsoft Access database](#), a Microsoft SQL Server 2000, 7.0, or 6.5 database, and other data sources.

[Microsoft Access and the OLE DB architecture](#)

OLE DB is a component database architecture that implements efficient network and internet access to many types of data sources, including relational data, mail files, flat files, and spreadsheets. In the OLE DB architecture, the application that accesses the data is called a data consumer (for example, Microsoft Access or a Microsoft Visual Basic program), and the program that enables native access to the data is called a database provider (for example, Microsoft OLE DB Provider for SQL Server or Microsoft Jet 4.0 OLE DB Provider).

[Defining connection information](#)

A data link is the connection information that a data consumer uses to access a data source through the OLE DB provider of that data source. In the Microsoft Access user interface, you use the **Data Link Properties** dialog box to define a data link for an OLE DB provider in the following ways:

- To connect to a Microsoft SQL Server database when you create a [Microsoft Access project](#) by using the **New** command on the **File** menu or by using the Database Wizard.
- To redefine the connection information for an Access project by using the **Connection** command on the **File** menu. You might do this to easily switch from a test to a production database or from a primary to a secondary server.
- To redefine the connection information of an existing [data access page](#) in [Page Design view](#) by right-clicking on the [field list](#) and clicking **Connection**. When you first create a data access page, Microsoft Access automatically defines the connection information for you by connecting to the currently open Access database or project.
- To connect a Web page that you opened from the **Open** dialog box (**File**

- menu, **Open**) from a null database window.
- To create a PivotTable list or modify the connection to a PivotTable list.

Each OLE DB provider defines specific connection information. For example, Microsoft OLE DB Provider for SQL Server requires a server name, server location, and a user name, and you may want to define additional information, such as a password or whether you want to use Microsoft Windows integrated security.

In general, to define connection information in the **Data Link Properties** dialog box:

- Click the **Provider** tab, select the OLE DB provider, and then click **Next** to display the **Connection** tab to enter specific connection information for that OLE DB provider. Note that depending on how you get to the **Data Link Properties** dialog box, the **Provider** tab may not be available. Access automatically defines the provider as the currently open Access database or SQL Server database connected to the open Access project.
- Click the **Advanced** tab to provide additional information, such as network settings and access permissions.
- Click the **All** tab to define initialization properties for that OLE DB provider.

Note If the Access project on the client computer is connected to a SQL Server 6.5 database running Microsoft Data Access Components (MDAC) 2.5, you must delete the generated entry for the Network Library property in the **All** tab.

The **Data Link Properties** dialog box Help file (msdasc.hlp) provides additional information on using the dialog box and defining connection information for each OLE DB provider.

 [OLE DB provider installation](#)

Microsoft Jet 4.0 OLE DB Provider and Microsoft OLE DB Provider for SQL Server are installed automatically when you install Microsoft Office 2003. Additional OLE DB providers are also installed, including OLE DB Provider for ODBC Drivers and OLE DB Provider for Oracle. To see a complete list of OLE DB providers installed on your computer, display the **Data Link Properties**

dialog box from a Data Link file, and then click the **Provider** tab.

When you or other users display a page using Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later and you have a Microsoft Office 2003 license on your computer, if necessary, the browser automatically downloads the necessary OLE DB providers and other related files by installing Microsoft Office Web Components.

[Programming OLE DB](#)

You can also programmatically access data through an OLE DB provider in Visual Basic for Applications in Microsoft [ActiveX Data Objects \(ADO\)](#). This is accomplished by defining a connection string in the **ConnectionString** property of the **Open** method of the **Connection** object to pass the connection information to the OLE DB provider.

[Show All](#)

Troubleshoot Access projects (ADP)

 [I can't find the Transact-SQL or Microsoft SQL Server error message help topics in Microsoft Access help.](#)

The Transact-SQL Help and SQL Server error message help topics are no longer available from the Access Help **Contents** Tab. For more information on Transact-SQL and SQL Server error messages, see the SQL Server Documentation.

 [I can't find the SQL Replication and Security menus.](#)

The **Database Security** and **Replication** menus previously available from the **Tools** menu of Microsoft Data Engine (MSDE) are no longer available on Microsoft SQL Server 2000 Desktop Engine. You can use SQL Server 2000 Enterprise Manager to replicate and help protect your database. You can also use the GRANT, REVOKE, and DENY Transact-SQL statements to help protect your database. For more information on SQL Server Enterprise Manage, SQL Server replication, and security, see the SQL Server Documentation.

 [I can't create an Access project shortcut on the Microsoft Windows Desktop.](#)

If you use the shortcut menu on the Windows Desktop (by right-clicking the desktop, pointing to **New**, and then clicking **Microsoft Access Application**) to try to create a new Access project (.adp) , you won't be able to. You can only create an Access database.

To create a shortcut to an Access project, locate the Access project in Windows Explorer, right click on the Acces project filename, Click **Create Shortcut**, and then drag the shortcut to the Windows Desktop.

 [Tip](#)

You can also drag a any database object in an Access project or Access database to the desktop.

 [I'm having problems backing up, restoring, or dropping a SQL Server database connected to my Access Project.](#)

Access cancels a backup, restore, or drop database operation if you don't have adequate permissions. Access cancels a backup operation if you don't have enough disk space to create the backup file. Make sure you have the correct permissions in SQL Server and adequate disk space before doing any of these operations.

You cannot backup, restore, or drop a SQL Server database running on a remote computer using an Access project. You can backup, restore, and drop a SQL Server 7.0 or later database or SQL Server 2000 Desktop Edition database on a local computer, or use SQL Server Enterprise Manager for these operations on a remote computer.

 [I'm having problems repairing my Access Project.](#)

When you repair an Access project file (.adp) (Point to **Database Utilities** on the **Tools** menu, and then click **Compact and Repair Project**), Access must open the file in Exclusive mode to perform the repair operation.

Make sure that all other users have closed the Access project before attempting to repair it.

Note The **Compact and Repair Project** command compacts and repairs an Access project, not a SQL Server database. To compact and repair a SQL Server database, use SQL Server Enterprise Manager. For more information on SQL Server Enterprise Manager, see the SQL Server documentation.

 [I'm having problems displaying international characters.](#)

- Your data may be stored in OEM format and the conversion setting is wrong. To determine whether data was stored in OEM format, you can use a query to display the contents of the tables you are working with. If extended characters appear incorrectly, the OEM conversion setting is probably wrong.

Close the Access project, change the setting in the SQL Server Client Network Utility program (under the **DB Library Options** tab), and then re-

open the Access project.

- You may see unexpected behavior if the system code page of client machines doesn't match the character set of the SQL Server. On the client, the system code page is the character set that is used by the language specified in the operating system; if you are running Windows 2000 or later, this is the Default Locale. On the server, the character set is specified during SQL Server installation.

In an Access project (*.adp), query results containing ANSI text fields are not displayed correctly if the system code page of the client computer does not match the character set of the SQL Server to which the client is connected. This problem does not occur for data that is stored as Unicode on the server.

 [I'm having problems creating or modifying a name for a database object.](#)

Valid database object names begin with a character or an underscore (_) and contain a combination of characters, numbers, and underscores. The maximum length for an object name stored in the database (queries, database diagrams, and tables) is 128 for SQL Server version 7.0 or later, and 30 for SQL Server version 6.5.

- The first character must be an underscore, or a letter as defined by Unicode Standard 2.0. The Unicode definition of letters includes Latin characters a-z and A-Z, in addition to letter characters from other languages.

In general, object names should not begin with an at sign (@), or pound sign (#). SQL Server uses a leading pound sign in an object name to indicate that the object is a temporary object. An identifier beginning with an at sign (@) character denotes a local variable or parameter. Some Transact-SQL functions have names that start with double at sign (@@) characters.

- Subsequent characters in an object name can be:
 - Letters as defined in the Unicode Standard 2.0.
 - Decimal numbers from either Basic Latin or other national scripts.
 - The at sign (@), dollar sign (\$), pound sign (#), or underscore (_) characters.

The object name should not contain Transact-SQL reserved keywords. Transact-SQL reserves both the uppercase and lowercase versions of reserved keywords. For a list of reserved keywords, see the SQL Server documentation.

Special characters, including the double-quote character ("), are not allowed.

In general, avoid names that contain periods.

[I can't change the database owner of my table, view, stored procedure, or user-defined function.](#)

In an Access project, you cannot change the database owner of an object using the database, table, or query designers. If you need to change the database owner of an object, use SQL Server Enterprise Manager. For more information on database object ownership and SQL Server Enterprise Manager, see the SQL Server documentation.

[I'm having problems opening or saving a database object.](#)

- You might not have permission to perform this operation. Contact the owner of the object or the database administrator to get permission to modify this object.
- The object might no longer exist in the database. Refresh the [database window](#) to find out if this object still exists in the database.
- You are not logged in as the database owner or as a user that is a member of the db_owner role.

Because you are not logged on as the system administrator, database owner, or a user that is a member of the db_owner role, you have limited privileges to the database. The privileges you have are determined by the permissions granted to your logon ID and the privileges granted to the roles that your logon ID is a member of.

- The view, stored procedure, or user-defined function is encrypted and can't be opened again.

Encryption helps prevent all users (regardless of database privileges), including yourself, from ever seeing the view, stored procedure, or user-defined function.

In Access 2000, you could encrypt a view from an Access project through the **Properties** dialog box of the View window. In Access 2002 or later, you can only encrypt views, stored procedures, or user-defined functions using Transact-SQL (by adding the WITH ENCRYPTION clause in the SQL pane) or by using SQL Server. For more information on encryption, see the SQL Server documentation.

Important Once you encrypt a view, stored procedure, or user-defined function, you can't change it. If you anticipate the need to revise it, save the SQL code in a text file in a controlled location before you encrypt it. When you need to revise it, you can do the following:

1. Delete the old encrypted view, stored procedure, or user-defined function.
2. Create a new view, stored procedure, or user-defined function with the same name.
3. Cut and paste the SQL from the saved text file into the SQL pane of the new view, stored procedure, or user-defined function.
4. Revise the new view, stored procedure, or user-defined function as desired.
5. Save the revised SQL in a new text file in a controlled location.
6. In the SQL pane, encrypt the new view, stored procedure, or user-defined function using the WITH ENCRYPTION clause.

 [Access deleted the wrong database object or a table has missing data.](#)

If your Access project is connected to Microsoft SQL Server or Microsoft 2000 Personal SQL Server, the sort order is set to "Dictionary order, case-insensitive," and Unicode collation case sensitivity is turned on, you can see unexpected behavior with database objects that have the same name but are spelled with all or some letters in a different case. For example you can accidentally delete the wrong table if you have three tables named "categories," "Categories," and "CATEGORIES." Access finds the first occurrence of the object name that matches the spelling but does not try to match the case because it is using a dictionary sort order that is case-insensitive. Therefore, the wrong object could be deleted.

Access uses the sort order to determine which object to delete or open. The default sort order of Microsoft SQL Server and Microsoft 2000 Personal SQL

Server is "Dictionary order, case-insensitive," meaning they ignore case when acting upon objects. Also, the default setting for Unicode Collation is case-insensitive. However, if during the setup of Microsoft SQL Server or Microsoft 2000 Personal SQL Server, the Unicode Collation case sensitivity is turned on, but the Sort Order is left at the default of "Dictionary order, case-insensitive," it allows objects with the same spelling but with different case to exist on the server.

When you install SQL Server or Microsoft 2000 Personal SQL Server make sure you select Custom Setup. In the **Character Set/Sort Order/Unicode Collation** dialog box, set the Sort Order to Dictionary order, case-insensitive and under **Unicode Collation**, make sure the **Case-insensitive** option is not selected.

Note This behavior does not occur if the server was set up with case-sensitive sort order as well as case-sensitive Unicode collation.

[I'm having problems using databases created with Microsoft Data Engine \(MSDE\) 1.0 under Microsoft SQL Server 2000 Desktop Engine](#)

If you uninstall Microsoft Data Engine (MSDE) 1.0, and then install SQL Server 2000 Desktop Engine, you can connect your Access project to the database file (.mdf) created by MSDE. However, when you connect this database file, Access converts it from SQL Server 7.0 to SQL Server 2000. Therefore, you will not be able to either convert them back to SQL Server 7.0 or use them with MSDE.

It's possible to have MSDE and SQL Server 2000 Desktop running on the same computer. You can do this by providing an instance name (MSOFFICE, for example) when you do a custom installation of SQL Server 2000 Desktop Engine and ensuring your Access project client computer has Microsoft Data Access Components (MDAC) version 2.6 installed. In this case, you can transfer a SQL Server 7.0 database (Point to **Database Utilities** on the **Tools** menu and then click **Transfer Database**) to a SQL Server 2000 database, but you cannot transfer a SQL Server 2000 database to a SQL Server 7.0 database.

[Show All](#)

Improve performance of an Access project

Depending on your computer's configuration and your working environment, there are several things you can do that may improve the performance of Microsoft Access or your [Access project](#).

[Improve the performance of an Access project](#)

In a client/server application, the general rules of thumb for reducing network traffic and optimizing application performance are:

- Do your data processing work on the server.
- Minimize roundtrips to the server to access the data.
- Limit the data you retrieve from the server.

[Understanding OLE DB and the client/server environment](#)

A [Microsoft Access project](#) uses OLE DB to connect and communicate with a Microsoft SQL Server database. Microsoft designed the [OLE DB](#) architecture to work efficiently in a client/server environment.

When you access data through a form or datasheet, OLE DB retrieves an [updateable snapshot recordset](#) from the SQL server database (up to the [maximum record limit](#)) and caches the data on the client—making one roundtrip to the server. Access retrieves the data asynchronously, which means that you can continue to interact with Access while the data is being loaded. When you work with the data on the form or datasheet, whether browsing, filtering, sorting, finding or updating data, you are working with the data that is cached on the client.

[Processing data on the server](#)

To process data on the server, you can use stored procedures, triggers, user-defined functions, and SQL SELECT statements. You can also sort data on the server before retrieving it.

Stored procedures are a precompiled collection of one or more SQL statements and optional control-of-flow statements that can accept parameters and are processed as a single unit by Microsoft SQL Server. A stored procedure resides in the SQL Server database. Because a stored procedure is compiled on the server when it is created, it executes faster than individual SQL statements. Because a stored procedure runs on the server, it takes advantage of the extra processing power of the server.

Triggers are a special kind of stored procedure that automatically execute when you update, insert, or delete data. Use triggers to enforce business rules and restrictions that are more complex than those defined with check [constraints](#). Unlike check constraints, triggers can reference columns in other tables. For example, a trigger can roll back updates that attempt to apply a discount (stored in a discounts table) to books (stored in a titles table) with a price of less than \$10.

User-defined functions combine the best features of stored procedures and views into a single query that you can nest, pass parameters to, sort, and return values. In many cases, user-defined functions are a better alternative to stored procedures because you can return a single table of data or scalar value, hide the business logic and details of how that data or scalar value was created from the user, and simplify the complexity of your SQL statement syntax.

Sort records on the server by using a stored procedure, user-defined function, or SQL statement stored in the **RecordSource** property of a form or report (which is stored in the Access project, but executes on the server). Sorting records on the server is another way to take advantage of the extra processing power of the server.



[Limiting data and property information retrieved from the server](#)

As much as possible, limit the amount of data your application retrieves from the SQL Server database. Use [views](#), stored procedures, user-defined functions, server filters, and SQL statement WHERE clauses to limit the data you or your users see in a form or datasheet. In general, avoid designing application

scenarios where users browse the database in unrestricted fashion.

You can use the [maximum record limit](#) button or the **MaxRecords** property to limit the data you or a user can see in a form or datasheet.

You can also use server filters to limit records before they are retrieved from the database on the server, and filters to further limit records on the client. You can define a server filter in the **ServerFilter** property of a form or report. You can also optimize the performance of lists of values in [Server Filter by Form](#) and [Filter by Form](#) windows by modifying the **FilterLookup** property to control the number of values displayed if the values are retrieved from either the server or client.

Disable automatic row fix-up if not required. After you update a record in a SQL Server 2000 database, Access normally resynchronizes the data to reflect the changed values. However, these operations require additional round trips to the server. You can disable row fix-up by setting the **PerformResync** property to **No**.

Disable pre-fetching of default values if not required. By default, when you display a form or datasheet, Access shows the default values in the new row of the datasheet and the form controls for new records on forms. This requires a round-trip to the server. You can disable the pre-fetching of default values by setting the **Fetch Defaults** property to **No**.

[Optimizing the SQL Server database](#)

The following are general guidelines for optimizing SQL Server database performance. For detailed information on improving SQL Server database performance, see the SQL Server documentation.

- Design tables without redundant data. A well-designed database is a prerequisite for fast data retrieval and updates.
- Enforce [referential integrity](#), which preserves the defined relationships between tables when you add, update, or delete the rows in those tables by defining primary key and foreign key constraints in the related tables.
- Choose an appropriate data type for fields. This can save space in your database and improve [join](#) operations. When defining a field, choose the smallest data type or field size that's appropriate for the data in the field.

- Create indexes for fields that you sort, join, or set criteria for. You can make dramatic improvements in the speed of a [query](#) by indexing fields that are on both sides of joins, and fields that are used to set criteria for the query. Indexes, however, can slow down the speed of record insertions, updates, and deletes.
- Use indexed views in SQL Server 2000 databases. Indexed views can greatly improve the performance of views found in data warehouse or other decision support applications where large amounts of data are aggregated into sums and averages. An indexed view forces the view's results to be stored in the database, so statements that reference the view will have substantially better response times. When using indexed views, you should be aware that modifications to the base data are automatically reflected in the view, which may cause additional maintenance overhead.
- Add a timestamp field to your tables to improve performance of record deletions and updates, especially if your table contains many fields. SQL Server uses a timestamp field to indicate that a record was changed (not when it was changed) by creating a unique value field, and then updating this field whenever a record is updated. SQL Server uses the value in timestamp fields to determine whether a record has been changed before updating it. In general, a timestamp field provides the best performance and reliability, especially in a multiuser environment. Without a timestamp field, SQL Server must check all the fields in the record to determine if the record has changed, and this check can slow performance.
- Periodically compact your SQL Server database to save disk space and reorganize indexes and data for faster access.

[Optimizing linked tables created in the Link Table Wizard](#)

If the SQL Server database you are linking to is on another SQL Server, create a linked server (select **Linked SQL** on the first page of the Link Table Wizard). SQL Server, in most cases, will optimize joining operations.

Create an OLE DB data source that uses Transact SQL functions (select **Transact SQL** on the first page of the Link Table Wizard) if you are linking to Excel, text, or HTML files because the performance is generally better than creating a linked server.

[Optimizing forms](#)

When you design a form, determine the data and functionality needed, and then delay asking the server for this data and functionality until the information is requested by the user. For example, create a form so that it doesn't require data to be retrieved from the server during the form-opening process. For example, you can add a [command button](#) to the form to load data on the form.

Use SQL UPDATE and DELETE statements with the appropriate WHERE clause to update or delete multiple records that meet the same criteria. This is more efficient than opening a [recordset](#) and performing the updates or deletions one record at a time.

Display fields that take a long time to retrieve data from the server, such as [text](#) and [image](#) fields, only when requested. You can use the following techniques:

- Place text and image fields on a subform that uses the same [record source](#). Access only retrieves the text or image values for the current record.
- Display the most important fields on a main form, and provide a button labeled "More Information" that opens another form containing other fields. Base the second form on a [query](#) that passes as a parameter the primary key field on the main form.
- If your application is data-entry intensive, such as an order entry system, create a specific form for adding records that has the **DataEntry** property set to **Yes**. When you open the form to enter new data, Access doesn't display any existing records. This saves time because Access doesn't have to retrieve all the records from the server.

If a form based on a stored procedure or user-defined function that is a parameter query, use the **Input Parameters** property to supply the parameter values.

[Optimizing reports](#)

Instead of printing a long report, break up reports into smaller subsets of data by using views, stored procedures, user-defined functions, and SQL statements as your [record source](#), or by using filters.

Use layout preview instead of print preview when you design your reports. You can take a quick look at the layout, which includes just a sample of the data in layout preview. When you are done working on the layout of the report, verify the data by opening the report in print preview.

You typically create and distribute reports on a schedule according to regular business cycles, such as weekly stock-level reminders, monthly sales reports, or quarterly financial statements. Create and distribute these reports as [report snapshots](#) in a batch file. Users can use the [Snapshot Viewer](#) to view and print the reports. When your users need up-to-the-minute reporting, use grouped data access pages.

If a report is based on a stored procedure or user-defined function that is a parameter query, use the **Input Parameters** property to supply the parameter values.

[Improve the performance of Microsoft Access and your system](#)

The following guidelines can help you optimize the performance of Microsoft Access on your computer:

- If you're using databases that other users don't need to share, install Microsoft Access and all your databases on your hard disk drive rather than on a network server.
- If you are the only person using a database, open the database for exclusive use; in the **Open** dialog box, click the arrow next to **Open** button, and then click **Open Exclusive**.
- Make more memory available by closing applications that you aren't using.
- Increase RAM on your computer. 40 MB of memory is recommended— 32 MB of memory plus an additional 8 MB of memory for Microsoft Access.
- Don't use any of your RAM for a RAM disk.
- Periodically delete unnecessary files and empty your Recycle Bin, compact your databases, and then defragment your hard disk with the Microsoft Windows Disk Defragmenter. To run the Disk Defragmenter, click the Windows **Start** button, point to **Programs**, point to **Accessories**, point to **System Tools**, and then click **Disk Defragmenter**.
- In most cases, the default virtual memory setting used by Windows should perform optimally. However, in some situations, adjusting virtual memory parameters can improve performance. If you've already tried deleting unnecessary files and you still have a performance problem, try changing the default virtual memory setting in the following cases:
 - You don't have much disk space available on the drive that is currently being used for virtual memory, and another local drive with space is available.

- Another local drive is available that is faster than the current drive (unless that disk is heavily used).

In these cases, you might get better performance by specifying a different drive for virtual memory.

You also might get better performance by specifying that the disk space available for virtual memory be at least 25 MB minus available RAM. For example, if your computer has 12 MB of RAM, you should specify at least 13 MB of virtual memory. You may want to specify more if you are running several large applications.

To change Windows virtual memory parameters, in Windows Control Panel, double-click the System icon, click the **Performance** tab, click **Virtual Memory**, and then click **Let Me Specify My Own Virtual Memory Settings**. Then specify a different hard disk, or enter a value in the **Minimum** box that is at least 25 minus your available RAM.

- If you have a wallpaper (full-screen background) bitmap on your Windows desktop, replace it with a solid color or pattern bitmap, or no bitmap at all.
- If you use a screen saver, use a blank screen saver or consider not using one at all.
- Eliminate disk compression software, or consider moving your databases to an uncompressed drive.
- To ensure optimal performance, use substitution fonts only if you need dual-font support to display all of your data.

[Improve filter performance](#)

If the lists in fields in the [Filter By Form](#) window or [Server Filter By Form](#) window take too long to display or aren't displaying values from the underlying table, you can change their performance. For both types of windows, you can prevent the lists from displaying the underlying table's field values, or you can change the record limit that determines if the list displays a field's values.

[Optimize Filter By Form performance for all tables, queries, and forms in the database or optimize Server Filter By Form performance for all forms](#)

For Filter By Form, these settings affect all tables and queries, and all text box

[controls](#) whose **FilterLookup** property is set to **Database Default**. For Server Filter By Form, these settings affect all text box controls whose **FilterLookup** property is set to **Database Default**.

1. On the **Tools** menu, click **Options**.
2. Click the **Edit/Find** tab, and then set the **Filter by form defaults** according to the performance that you want to achieve.
 - If the list of values takes too long to display, you can prevent the list from retrieving the unique values for each field by clearing the **Records in local snapshot** check box, which affects the Filter By Form window, and the **Records at server** check box, which affects the Server Filter By Form window.
 - If lists aren't displaying the values from fields, check under **Show list of values in** to make sure that the appropriate boxes are selected, or try increasing the number in the **Don't display lists where more than this number of records read** box so that it's greater than or equal to the maximum number of unique values in any field in the underlying table or other record source.

 [Optimize Filter By Form or Server Filter By Form performance for a specific form](#)

For each text box [control](#) on a form, you can specify whether the values stored in the field appear on its list in the Filter By Form window or in the Server Filter By Form window.

1. Open a form in [Design view](#).
2. Display the property sheet for any text box control.
3. In the **FilterLookup** property box, set the property to the setting that will achieve the performance you want.
 - If the list takes too long to display in the Filter By Form window or Server Filter By Form window, either set the **FilterLookup** property to **Never**, or set it to **Database Default** and then make the appropriate changes in the **Options** dialog box (**Tools** menu).
 - If the list in the Filter By Form window or Server Filter By Form window isn't displaying the values that are stored in that field, or it's displaying only some of the values, make sure the **FilterLookup** property isn't set to **Never**. If the **FilterLookup** property is set to **Database Default** or **Always**, make the appropriate changes in the

Options dialog box (**Tools** menu).

[Improve find and replace performance](#)

[What is an index?](#)

You can use an index to gain fast access to specific information in a database table. An index is a structure that orders the values of one or more columns in a database table, for example the last name (lname) column of the employee table. If you were looking for a specific employee by his or her last name, the index would help you get that information faster than if you had to search all the rows in the table.

The index provides pointers to the data values stored in specified columns of the table, and then orders those pointers according to the sort order you specify. The database uses the index much as you use an index in a book: it searches the index to find a particular value and then follows the pointer to the row containing that value.

In database diagrams, you can create, edit, or delete each type of index in the **Indexes/Keys** property page for a selected table. An index is saved in the database when you save the table that it is attached to, or when you save the database diagram in which that table appears.

As a general rule, you should create an index on a table only if the data in the indexed columns will be queried frequently. Indexes take up disk space and slow the adding, deleting, and updating of rows. In most situations, the speed advantages of indexes for data retrieval greatly outweigh these disadvantages. However, if your application updates data very frequently or if you have disk space constraints, you might want to limit the number of indexes.

Before creating an index, you must determine what columns to use and what type of index to create.

[Index columns](#)

You can create indexes based on a single column or on multiple columns in a database table. Multiple-column indexes enable you to distinguish between rows in which one column may have the same value.

Indexes are also helpful if you often search or sort by two or more columns at a time. For example, if you often set criteria for last name and first name columns in the same query, it makes sense to create a multiple-column index on those two columns.

To determine the usefulness of an index:

- Examine the WHERE and JOIN clauses of your queries. Each column included in either clause is a possible candidate for an index.
- Experiment with the new index to examine its effect on the performance of running queries.
- Consider the number of indexes already created on your table. It is best to avoid a large number of indexes on a single table.
- Examine the definitions of the indexes already created on your table. It is best to avoid overlapping indexes that contain shared columns.
- Examine the number of unique data values in a column and compare that number with the number of rows in the table. The result is the selectivity of that column, which can help you decide if a column is a candidate for an index and, if so, what type of index.

[Types of indexes](#)

Depending on the functionality of your database, you can create three types of indexes - unique, primary key, and clustered - in Database Designer.

[Tip](#)

Although a unique index will help locate information, for the best performance results it is recommended that you use primary key or unique constraints instead.

[Unique Index](#)

A unique index is one in which no two rows are permitted to have the same index value.

Most databases prevent you from saving a table with a newly created unique index when there are duplicate key values in the existing data. Your database may also prevent the addition of new data that would create duplicate key values in the table. For example, if you create a unique index on the employee's last

name (lname) in the employee table, then no two employees can share the same last name.

[Primary Key Index](#)

A database table often has a column or combination of columns whose value uniquely identifies each row in the table. This column is called the primary key of the table.

Defining a primary key for a table in a database diagram automatically creates a primary key index that is a specific type of unique index. This index requires each value in the primary key to be unique. It also permits fast access to data when you use the primary key index in queries.

[Clustered Index](#)

In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the key values. A table can contain only one clustered index.

If an index is not clustered, the physical order of the rows in the table does not match the logical order of the key values. A clustered index usually provides faster access to data than does a nonclustered index.

[Improve form and subform performance](#)

There are several things you can do to make your forms run faster.

[General tips](#)

- Avoid overlapping controls.
- Don't sort records in an underlying [query](#) unless record order is important, especially with multiple-table queries.
- If the underlying [record source](#) includes many records and you want to use the form primarily to enter new records, set the **DataEntry** property of the form to **Yes** so that the form opens to a blank record. If you open a form with all records showing, Microsoft Access has to read each record before it can display the blank record at the end of the recordset.
- Close forms that aren't being used.

[Tips for forms that contain pictures and objects](#)

- Use bitmaps and other graphic objects sparingly.
- Convert unbound object frames that display graphics to image controls.
- Use black-and-white rather than color bitmaps.

[Tips for forms that contain code](#)

A form without a [form module](#) loads more quickly and occupies less disk space. If a form or controls on the form don't use [event procedures](#), the form doesn't require a form module.

- Eliminate code from forms that don't need it by setting the **HasModule** form property to **No**. The **HasModule** property specifies whether the form has a form module.
- Instead of using command buttons with event procedures, you can use command buttons with [macros](#), or [hyperlinks](#).

Notes

- If a form currently contains event procedures, and you decide to eliminate all event procedure code from that form, you must set the **HasModule** property to **No** to completely remove the form module.
- You can still use code with a form that has no form module by calling **Function** procedures from a standard module using an [expression](#). (You can't use **Sub** procedures, because they can't be called using an expression.) To do this, define a **Function** procedure in a standard module and then call that function from an [event property](#) of the form or a control. For example, to use a command button to call a function to open a form, add an expression to the **OnClick** property of the command button like this:
=OpenDataEntry().

[Tips for subforms](#)

- Base subforms on queries rather than on tables, and include only fields from the [record source](#) that are absolutely necessary. Extra fields can decrease subform performance.
- [Index](#) all the fields in the subform that are linked to the [main form](#).

- Index any subform fields that are used for criteria.
- Set the subform's **AllowEdits**, **AllowAdditions**, and **AllowDeletions** properties to **No** if the records in the subform aren't going to be edited. Or set the **RecordsetType** property to **Snapshot**.

[Improve list box and combo box performance](#)

There are several things you can do to make your list boxes, combo boxes, and [drop-down list boxes](#) run faster:

- Base the list box, combo box, or drop-down list box on a saved [query](#) instead of an SQL statement. On a form, if you use a wizard to create a list box or combo box, Microsoft Access automatically sets the **RowSource** property of the control to an SQL statement. To change the **RowSource** property to a saved query, click the **Build** button  next to the **RowSource** property box. With the **Query Builder** window open, click **Save** on the **File** menu, and type a name for the query. When you close the Query Builder window, click **Yes** when Microsoft Access asks if you want to update the property.
- Base the drop-down list box on a saved query instead of a table. On a data access page, if you use a wizard to create a drop-down list box, Access automatically sets the **ListRowSource** property of the control to the table you specified in the wizard. To retrieve rows from a query instead of a table, select the query name from the **ListRowSource** property box.
- In the query specified in the **RowSource** or **ListRowSource** property box, include only fields that are absolutely necessary. Extra fields can decrease performance.
- [Index](#) both the first field displayed in the combo box, list box, or drop-down list box and the bound field (if the fields are different).
- In combo boxes on a form, set the **AutoExpand** property to **No** if you don't need the fill-in-as-you-type feature.
- On a form, if the **AutoExpand** property for a combo box is set to **Yes**, the first displayed field should have a **Text** data type instead of a Number data type. In order to find a match in the list, Microsoft Access converts a numeric value to text. If the data type is **Text**, Microsoft Access doesn't have to do this conversion.
- On a form, if the bound field in a lookup combo box is not the displayed field:
 - Don't use [expressions](#) for the bound field or the displayed field.

- Don't use restrictions in the row source.
- Use single-table (or query) row sources rather than multiple-table row sources, if possible.
- Don't create list boxes, combo boxes or drop-down list boxes based on data in a linked table if the data won't change. It's better to import the data into your database in this case.

Note The Query Builder is not available in a data access page. Wizards are not available in a stand-alone data access page, or when you point a data access page to a database other than the one currently open. In both cases, you will have to create a list box or drop-down list box on your own.

[Improve report and subreport performance](#)

There are several things you can do to make your reports print faster.

[General tips](#)

- Avoid overlapping controls.
- Avoid sorting and grouping on expressions.
- [Index](#) fields you sort or group on.
- Avoid using [domain aggregate functions](#). Include the field in the report's underlying query or use a [subreport](#).
- Make sure the report's underlying query is optimized.
- Use the **HasData** property or NoData event to determine if a report is bound to an empty recordset.

[Tips for reports that contain pictures and objects](#)

- Use bitmaps and other graphic objects sparingly.
- Convert [unbound object frames](#) that display graphics to [image controls](#).
- Use black-and-white rather than color bitmaps.

[Tips for subreports](#)

- Base subreports on queries rather than tables, and include only fields from the record source that are absolutely necessary. Extra fields can decrease subreport performance.

- Index all the fields in the subreport that are linked to the main report.
- Index any subreport fields used for [criteria](#).

[Improve data access page performance](#)

You can do several things to make your data access pages load faster in [Page view](#) or in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later.

[General tips](#)

- When using a data access page within Microsoft Access, close all database objects that aren't being used. When using a data access page in Internet Explorer, close all windows that aren't being used.
- On data access pages that are used for data entry, set the **DataEntry** property of the page to **True** so that the page opens to a blank record.
- Index any fields that you sort, group, or filter on.
- Bind a data access page to an XML data file, so the page could get its data from a file located on the local client instead of having to access the database server.

[Controls](#)

- Avoid overlapping controls.
- Use bound span controls instead of text boxes to display any data that isn't updateable— for example, to display autonumbered values, values derived from expressions, and hyperlinks.

Security Use bound span controls or hyperlink controls with caution. Unauthorized users may be able to insert harmful HTML text into these controls. Processing this text without validation or modification could result in the loss of personal information or other data. To prevent the HTML text contained in a control from being rendered, encode the contents of the control.

[Pictures and objects](#)

- Use bitmaps and other graphic objects sparingly. Use black-and-white rather than color bitmaps.

[Grouped pages](#)

- Make sure the **ExpandedByDefault** property is set to **False** for the highest group level. Setting it to **False** for lower group levels will speed up other interactions after the page is opened.
- Make sure the **DataPageSize** group level property is set to a low rather than a high number, or is set to **All**. The lower the number, the faster the records are displayed.
- On a page that contains records with a one-to-many relationship, group records by table rather than by a field or expression.
- On a page that is based on tables with a one-to-many relationship, bind each section to a table rather than to a query. Microsoft Access retrieves the records from a table only as it needs to display them on the page. With a query, Access retrieves all the records before it displays any records on the page. Use a query only when you want to limit the data used on the page, calculate aggregate values, or perform a task that's possible only in a query.

[Read about making your Access project an ADE file to improve performance](#)

If your [Microsoft Access project \(.adp\)](#) file contains [Visual Basic for Applications \(VBA\)](#) code, saving your Access project as an [.ade](#) file compiles all modules, removes all editable source code, and compacts the destination Access project file. Your VBA code will continue to run, but it cannot be viewed or edited.

Your Access project file will continue to function normally— you can still update data and run reports. Additionally, the size of your Access project will be reduced due to the removal of the code, so memory usage is optimized, which will improve performance.

Saving your Access project file as an [.ade](#) file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in [Design view](#).
- Adding, deleting, or changing references to [object libraries](#) or databases.
- Changing code— an [.ade](#) file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, data access pages, views, stored procedures, database diagrams, and macros can be imported from or exported to non-[.ade](#) Access projects.

[Considerations if you need to modify the design of forms, reports, or modules](#)

Be sure to save a copy of your original Access project file. If you need to modify the design of forms, reports, or modules in an Access project file that you have saved as an .ade file, you must modify the original Access project, and then resave it as an .ade file.

You won't be able to open, convert, or run code in an .ade file in future versions of Access. The only way to convert an ade file to a future version will be to open the original Access project that the .ade file was created from, convert it to the later version of Access, and then save the converted Access project file as an .ade file.

[Considerations before saving your Access project file as an .ade file](#)

Some restrictions may prevent you from saving your Access project file as an .ade file:

- You must have password access to the VBA code if a password is defined.
- If your Access project references another Access project or add-in, you must save all Access projects and add-ins in the chain of references as .ade or .mde files.

[About references and .ade files](#)

If you try to create an .ade file from an Access project file or an add-in that references another Access project or add-in, Access displays an error message and doesn't let you complete the operation. To save an Access project that references another Access project as an .ade file, you must save all projects in the chain of references as .ade or .mde files, starting from the first project referenced. After saving the first project as an .ade or .mde file, you must then update the reference in the next project to point to the new file before saving it as an .ade file, and so on.

For example, if Project1.adp references Project2.adp, which references Project3.mda, you would proceed as follows:

1. Save Project3.mda as Project3.mde.

2. Open Project 2.adp and change its reference to point to the new Project3.mde.
3. Save Project2.adp as Project2.ade.
4. Open Project1.adp and change its reference to point to the new Project2.ade.
5. Save Project1.adp as Project1.ade.

[Read about compacting an Access project](#)

Microsoft Access combines compacting and repairing an [Access file](#) into a single process.

[Compacting an Access file](#)

If you delete data or objects in an [Access database](#), or if you delete objects in an [Access project](#), the file can become fragmented and use disk space inefficiently. Compacting the Access file makes a copy of the file and rearranges how the file is stored on your disk. Compacting a previous-version Access database won't convert it to Access 2002 - 2003 format.

Compacting optimizes the performance of both Access databases and Access projects. However, in an Access project, compacting does not affect [database objects](#)— such as tables or views— that reside in the [Microsoft SQL Server database](#) instead of in the Access project itself.

Compacting has no effect on autonumbering in an Access project. However, in an Access database, if you have deleted records from the end of a table that has an [AutoNumber field](#), compacting the database resets the AutoNumber value; the AutoNumber value of the next record you add will be one greater than the AutoNumber value of the last undeleted record in the table.

[Repairing an Access file](#)

In most cases, Microsoft Access detects whether an Access file is damaged when you try to open it and gives you the option to repair it at that time. If the current Access file contains a reference to another Access file that is corrupt, Access does not attempt to repair the other file. In some situations, Access may not detect that a file is damaged. If an Access file behaves unpredictably, compact and repair it.

Access can repair:

- Corruption in a table in an [Access database](#).
- Missing information about the structure of an Access file's Visual Basic for Applications (VBA) project.
- Corruption in a form, report, or module.
- Missing information that Access needs to open a particular form, report, or module.

To help prevent corruption of Access files, follow these guidelines:

- Compact and repair Access files regularly. You can specify that Access compact a file automatically when you close it.
- Back up your Access files regularly.
- Avoid quitting Access unexpectedly. For example, don't quit Access suddenly by turning off your computer.
- If you are experiencing network problems, avoid using a shared Access database located on a network server until the problems are resolved. If possible, move the Access database to a computer where you can access it locally instead of over the network.

[Repairing files after Access shuts down unexpectedly](#)

If a serious problem occurs and causes Microsoft Access to shut down, Access restarts, creates a copy of the file that was open, and names it *filename_Backup.mdb* or *filename_Backup.adp*, where *filename* is the name of the original file. If *filename_Backup.mdb* or *filename_Backup.adp* already exists, Access asks you to specify the name of the file. Access then attempts to compact and repair the original file.

[Show All](#)

Open an Access project

Some of the content in this topic may not be applicable to some languages.

1. On the **File** menu, click **Open** .
2. Click a shortcut on the left side of the **Open** dialog box, or, in the **Look in** box, click the drive or folder that contains the [Microsoft Access project](#) (.adp) that you want.
3. In the folder list, double-click folders until you open the folder that contains the Access project.

If you can't find the Access project that you want to open, click **Tools** in the upper-right corner of the **Open** dialog box, and then click **Search**. In the **Search** dialog box, enter additional search criteria.

4. Click the Access project that you want to open, and then do one of the following:
 - To open the Access project, click **Open**.
 - To open the Access project for read-only access, click the **Open** arrow, and then click **Open Read-Only**.

You can open an Access project only in [exclusive](#) mode. If you attempt to open an Access project while another user has it open, Access prompts you to open a read-only copy of the file.

Note When you open an Access project in read-only mode, you can still add or modify data, and create or modify tables, views, database diagrams, or stored procedures, because the data and these objects reside in the Microsoft SQL Server. However, you can't create or modify forms, reports, macros, or modules, because these objects reside in the Access project.

[Tips](#)

- To view a list of shortcuts to Access projects that you have opened previously, click **History** on the left side of the **Open** dialog box.
- You can also click one of the file names located at the bottom of the **File** menu. If the list of recently used files isn't displayed, click **Options** on the

Tools menu, click the **General** tab, and then select the **Recently used file list** check box.

[Show All](#)

Create an Access project

Some of the content in this topic may not be applicable to some languages.

You can create a [Microsoft Access project](#) and connect it to a [Microsoft SQL Server](#) database.

1. On the **File** menu, click **New**.
2. Do one of the following:

[Create a Microsoft Access project and connect it to an existing Microsoft SQL Server database](#)

1. In the **New File** task pane, under **New**, click **Project (Existing Data)**.
2. In the **File New Database** dialog box, click a location in the **Save in** box, and type a name in the **File name** box; make sure that **Microsoft Access Projects** appears in the **Save as type** box, and then click **Create**.
3. On the **Connection** tab in the **Data Link Properties** dialog box, enter a server name and the information necessary to log on to the server, and then select the database on the server. For additional information, click **Help** in the **Data Link Properties** dialog box.

[Create a SQL Server database and connect an Access project to it](#)

1. In the **New File** task pane, under **New**, click **Project (New Data)**.
2. In the **File New Database** dialog box, click a location in the **Save in** box, and type a name in the **File name** box; make sure that **Microsoft Access Projects** appears in the **Save as type** box, and then click **Create**.
3. In the Microsoft SQL Server Database Wizard, type the name of the server you would like to use, the logon ID and password of an account with CREATE DATABASE privileges on the server, and the name of the new SQL Server database.

4. Click **Next**.
5. If you are creating a Microsoft SQL Server 6.5 database, enter the database device and size, the transaction log device and size, the database size, and the log size. You don't need to specify devices and sizes for Microsoft SQL Server 7.0 or later.
6. Click **Finish**.

[Create an Access project that is not connected to a Microsoft SQL Server database](#)

1. In the **New File** task pane, under **New**, click **Project (Existing Data)**.
2. In the **File New Database** dialog box, click a location in the **Save in** box, and then type a name in the **File name** box; make sure that **Microsoft Access Projects** appears in the **Save as type** box, and then click **Create**.
3. In the **Data Link Properties** dialog box, click **Cancel**.

[Show All](#)

Connect an Access project to a Microsoft SQL Server database (ADP)

Some of the content in this topic may not be applicable to some languages.

You can connect a [Microsoft Access project](#) to a [Microsoft SQL Server](#) database. If the Access project is already connected to a Microsoft SQL Server, you can change that connection.

1. Open the Access project that you want to connect to a Microsoft SQL Server.
2. On the **File** menu, click **Connection**.
3. On the **Connection** tab in the **Data Link Properties** dialog box, enter a server name and the information necessary to log on to the server, and then select the database on the server. For additional information, click **Help** in the **Data Link Properties** dialog box.

[Show All](#)

Link tables in an Access project by using the Link Table Wizard (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

In a [Microsoft Access project \(.adp\)](#) connected to a Microsoft SQL Server 7.0 or later database, you can use the Link Table Wizard to link to one or more tables in another SQL database, an Access database, or other OLE DB and [ODBC data sources](#).

Note With a local installation of Microsoft SQL Server or the Microsoft SQL Server 2000 Desktop Engine, you can use the Link Table Wizard to link to data in the following applications: Microsoft Access databases .mdb (all versions), Microsoft Access projects .adp (all versions), dBASE (versions 3, 4 and 5), Paradox (versions 3.x, 4.x, 5.x, and 7.x), Microsoft Excel (version 3.0 and above), delimited text files (using the system delimiter as set in **Regional Options** in the Windows Control Panel), and HTML. Without a local SQL Server installation, you can only link to other SQL Server tables.

1. Open an Access project, or switch to the [Database window](#) for the open Access project.
2. On the **File** menu, point to **Get External Data**, and then click **Link Tables**.
3. In the first screen of the **Link Table Wizard**, select the type of link you want to create, either **Linked SQL** or **Transact SQL**, and then click **Next**.

If you select **Linked SQL**, Access creates a linked server which provides the most functionality, including the ability to update data if the OLE DB data source allows, and permanently stores the data source connection information in the [Microsoft SQL Server database](#). Also, if the SQL Server database you are linking to is on another SQL Server, in most cases, SQL Server will optimize join operations. However, you must be a member of either the sysadmin or setupadmin roles.

If you select **Transact SQL**, Access creates an OLE DB data source that uses Transact SQL functions to create an ad hoc read-only connection to the

external data source. Access uses either the the OPENDATASOURCE (SQL Server 2000) or OPENROWSET (SQL Server 7.0) Transact SQL functions. If you are linking to spreadsheet, text, or HTML files, consider using this option because the performance is better than creating a linked server.

4. In the **Select Data Source** dialog box, do one of the following:

[Link to an existing data source](#)

1. Select the data source.
2. Click **Open**.

[Create a new data source](#)

1. Click **New Source**.
2. In the **Data Connection Wizard** dialog box, click each table that you want to link.
3. Follow the directions in the wizard dialog box.

Access creates a connection file with the information you specified.

4. In the **Select Data Source** dialog box, click **Open**.
5. In the second screen of **Link Table Wizard**, select the tables you want to link to, and then click **Next**.
6. Click **Finish** to complete the wizard.

The wizard creates a [view](#) of each linked table.

[Show All](#)

Show SQL Server properties (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

1. Point to the **View** menu and click **Server Properties**.
2. The **Server Properties** dialog box is displayed with following information:
 - The database product (Microsoft SQL Server).
 - The database version number (8.0, 7.0, or 6.5).

Note 8.0 refers to SQL Server version 2000.

- The name of the database server.
- The current [SQL Server database](#) name that the [Microsoft Access project](#) is connected to.

Note The current SQL Server database name, along with the file format (Access 2000 or Access 2002 - 2003), is also displayed in the [database window title bar](#) to the right of the Access project file name.

[Show All](#)

Drop a database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can drop a database in Microsoft SQL Server 2000 Desktop Engine from a Microsoft Access project. To drop a database in SQL Server, you can use the DROP DATABASE Transact-SQL statement, or use SQL Server Enterprise Manager. For more information on the DROP DATABASE statement and SQL Server Enterprise Manager, see the SQL Server documentation.

Note Dropping a database permanently deletes the database. To drop a database, you must disconnect all users from the database and close all windows except the [Database window](#).

1. Open an Access project connected to SQL Server 2000 Desktop Engine.
2. Point to **Database Utilities** on the **Tools** menu and then click **Drop SQL Database**.

Microsoft Access confirms the operation. Click **Yes** to drop the database.

[Show All](#)

Create a lookup column (ADP)

In a [Microsoft Access project](#) connected to a Microsoft SQL Server 2000 database, you can add a lookup column. A lookup column displays values in another table's column that correspond to the lookup column's value. For example, a CustomerID column in an Orders table that is a [foreign key](#) to a Customers table. In this case, you want to display information about the customer, such as the customer name and address, rather than the number in the CustomerID column.

1. In the Database window, click **Tables**  under **Objects**, click the table you want to open, and then click **Design** on the Database window toolbar.
2. Select an ID column, such as the foreign key column, and then click the **Lookup** tab.
3. Define the lookup column by setting the following properties:
 - In the **Display Control**, select the type of control you want to use for the lookup column in a form.
 - Select **Text Box** to create a [text box](#).
 - Select **List Box** to create a [list box](#) and make the next six properties available.
 - Select **Combo Box** to create a [combo box](#) and make all subsequent properties available.
 - In the **Row Source Type**, enter the type of row source (**Tables/Views/Functions, Value List, or Field List**).
 - In the **Row Source** property, enter the name of the row source:
 - If **Row Source Type** is set to **Tables/Views/Functions**, select a table, view, or function from the drop-down list box.

Note Although stored procedures are not listed, you can type the name of a single-statement stored procedure that does not modify data.

You can also click the **Build** button  to the right and use the Query Builder to create an SQL SELECT statement.

- If **Row Source Type** is set to **Value List**, enter the list of values, separated by semicolons, for example, "Small;Medium;Large"

(without quotes).

- If **Row Source Type** is set to **Field List**, enter the list of fields from the query or SQL statement specified by the **Row Source Type** setting that you want used in the list box or combo box.
 - In the **Bound Column** property, enter the number of the column you want bound to the underlying field for a bound multicolumn list box or combo box. This number is an offset: the first column is zero, the second column is 1, the third column is 2, and so on.
 - In the **Column Count** property, enter the number of columns you want displayed in the list box or combo box.
 - In the **Column Heads** property, select **Yes** if you want to display column headers.
 - In the **Column Widths** property, enter the width of each column in inches, separated by semicolons, for example, 2;3. To hide the bound column from the display, use a width value of zero, for example 0;2.
 - In the **List Rows** property, enter the maximum number of rows to display in the list box portion of a combo box.
 - In the **List Width** property, enter the width of the list box portion of a combo box in inches.
 - In the **Limit To List**, property, select **Yes** to restrict a combo box to displaying only values in the list. Select **No** to allow a user to enter text that doesn't match a value in the list.
4. Click **Save**  to save changes and switch to Datasheet view to verify your results.

Note You can add more than one lookup column to a table.

[Show All](#)

Create an input mask (ADP)

In a [Microsoft Access project](#), you can create an [input mask](#) in a table, query, or form.



[Create an input mask for a field in table Design view](#)

1. Open the table in Table Designer or the query in Query Designer.
2. Click the **Columns** tab in the Table Designer, or right click the in the Query Designer, click **Properties**, and then click the **Columns** tab.
3. Do one of the following:
 - Click the **Build** button  next to the **Input Mask** property to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.
 - Type the input mask definition in the **Input Mask** property field.



[Create an input mask for a control on a form](#)

1. Open a form in [Design view](#).
2. Select a text box or a combo box, and then click **Properties**  on the toolbar to open the control's property sheet.
3. In the **Input Mask** property box, do one of the following:
 - Click the **Build** button  in the **Input Mask** property box to start the Input Mask Wizard, and then follow the instructions in the wizard dialog boxes.
 - Type the input mask definition in the **Input Mask** property field.

[Show All](#)

Back up and restore a database or Access project (ADP)

You can back up and restore a database in Microsoft SQL Server 2000 Desktop Engine from a [Microsoft Access project](#). You should backup a database frequently and regularly to help protect your data from hardware failure or accidental loss. You can then restore the data from the backup you created at the time it was created. To backup and restore a database in SQL Server, use SQL Server Enterprise Manager. For more information on backing up and restoring databases and SQL Server Enterprise Manager, see the SQL Server documentation.

You can also back up and restore an Access project (.adp), which is a file separate from (but connected to) the SQL Server database.

[Back up and restore a SQL Server database](#)

[Back up a SQL Server database](#)

1. Open an Access project connected to SQL Server 2000 Desktop Engine.
2. Point to **Database Utilities** on the **Tools** menu and then click **Back Up SQL Database**.

The **Save Backup As** dialog box is displayed and prompts you for a backup file name (.dat).

3. Type a name for the backup file in the **File name** box, or use the default name entered by Access.

[Restore a SQL Server database](#)

Note To restore a database, you must disconnect all users from the database and close all windows except the [Database window](#).

1. Open an Access project connected to SQL Server 2000 Desktop Engine.

2. Point to **Database Utilities** on the **Tools** menu and then click **Restore SQL Database**.

The **Restore** dialog box is displayed.

3. Type the name of the file in the **File name** box or locate the backup file (.dat) in the **Restore** dialog box.

[Back up and restore an Access project](#)

[Back up an Access project](#)

1. Save and close all objects in the Access project.
2. On the **File** menu, click **Back Up Project**.
3. In the **Save Backup As** dialog box, specify the name and location of the backup copy.

[Restore an Access project from a backup copy](#)

- Depending on what method was used originally to produce the backup copy, use My Computer, the Windows 2000 or later Backup and Recovery Tools, or other backup software to copy the backup Access project to your database folder.

Caution If the existing Access project in the database folder and the backup copy have the same name, restoring the backup copy may replace the existing file. If you want to save the existing file, rename it before you copy the backup Access project.

Note You can back up individual [database objects](#) by creating a blank Access project and then importing the objects you want from the original database.

[Show All](#)

Create a column to store hyperlinks (ADP)

In a [Microsoft Access project](#) (.adp) connected to a Microsoft SQL Server 2000 database, you can create a character string field that contains a hyperlink.

1. In the Database window, click **Tables**  under **Objects**, click the table you want to open, and then click **Design** on the Database window toolbar.
2. In the first blank row, click in the **Column Name** column and type a unique name for the field.
3. In the **Data Type** column, select [char](#), [nchar](#), [nvarchar](#), or [varchar](#).
4. In the **Length** column, set an appropriate value (in number of characters) for the hyperlink addresses that you'll store.
5. In the column properties, under the **Columns** tab, scroll down to the **Hyperlink** property, and set the value to **Yes**.
6. Click **Save** to save the table definition.

Note If you don't want to change the table definition, you can still create a hyperlink on a form or report control containing a character string data type field by setting the **Is Hyperlink** property to **Yes**.

[Show All](#)

About filters (ADP)

The type of filter that you create depends on whether you want to filter records at the server before they are retrieved from the database, or locally after they are retrieved. It also depends on the type of object for which you're creating the filter.

[Ways to filter records before they are retrieved from the database](#)

For forms and reports, there are a number of ways that you can filter records at the server before they are retrieved from the database. By filtering records at the server, you can improve the form or report's performance. Besides basing the form or report on a [query](#) that includes [criteria](#) (search conditions) to limit the records, you can:

Define a specific filter in the **ServerFilter** property of the form or report. When you or others open the form or report, it contains only the records that meet the criteria that you specified in the **ServerFilter** property. You can't change the filter unless you change the property or use the filtering techniques described in the remainder of this help topic to limit the records once they are local.

Prompt for a different filter each time you open a form in [Form view](#) or [Datasheet view](#). You do this by setting the form's **ServerFilterByForm** property to **Yes**. Then, when you switch to Form view or Datasheet view, Microsoft Access displays the [Server Filter By Form](#) window, which is a copy of the form (or datasheet) that you are filtering, but without records. In the form, you can specify different criteria by choosing your search values from drop-down lists in each field. When you apply the filter, Microsoft Access sets the **ServerFilter** property to the filter you defined in the Server Filter By Form window.

Note You can't use the Server Filter By Form window with a form bound to a [stored procedure](#). Instead, consider using parameters in the stored procedure, and then setting the form's **InputParameters** property to bind the parameters to the form.

[Ways to filter records after they are retrieved from the database](#)

For forms and datasheets, you can filter records after they've already been retrieved. This is true even for forms that were initially filtered at the server. Once the records are displayed, you can use these local filtering techniques to limit the records:

Filter By Form displays a copy of the form (or datasheet) that you are filtering, but without records. You can specify different [criteria](#) each time you use the form or datasheet by choosing your search values from drop-downlists in each field or by specifying an expression. There's no need to scroll through all the records in a form or datasheet, and you can specify multiple criteria at once.

Filter By Selection finds records with a field containing all or part of a value you selected in that same field on a form, subform, or datasheet. If you want to further limit the records, you can continue selecting other values or partial values. Each time you apply the filter, it uses the previous values that you selected and any subsequent values to determine the filtered set of records.

Filter For Input displays an area on the shortcut menu where you can type the exact value that you're searching for in the current field or the expression whose result you want to use as your criteria.

[Ways to filter records for a data access page](#)

[Ways to filter records before they are retrieved from the database](#)

Filtering records before they are retrieved from the database can improve data access page performance because there are fewer records to download from the underlying database.

- You can base your page on a [query](#) that includes criteria (search conditions) to limit the records.
- In a [Microsoft Access database](#) you can define a server-side filter in the **ServerFilter** property of the **RecordsetDef** object, or by using the **Filter** property in Visual Basic for Applications (VBA) or Microsoft Visual Basic Scripting Edition (VBScript). In a [Microsoft Access project](#) you can define a server-side filter in the **ServerFilter** property in VBA or VBScript.
- On a grouped data access page, you can create a filter that retrieves data for

a specific group, such as orders for a specific customer. The data access page won't display any records until you choose the group from a drop-down list in [Page view](#) or Microsoft Internet Explorer. Once you choose the group, the filter is applied and only those records are retrieved.

On an ungrouped data access page, you can create a filter by designating a field as the filter field. When you choose a value from the filter field, all records that contain the selected value are retrieved.

- In [Design view](#), you can add a server-side filter to a hyperlink on a page. Then, when you click the link in Page view or in the browser, only the records specified in the filter are displayed on the page you jump to.

[Ways to filter records after they are retrieved from the database](#)

Use the [Filter By Selection](#) technique in [Page view](#) or the browser. With this type of filter, you select in a field the value that you want the filtered records to contain.

[About filtering data in a PivotTable or PivotChart view](#)

[About filtering data in a PivotTable view](#)

You can use the filter feature to find specific data values or all data that matches a value.

[Filtering techniques](#)

Filtering a field (Autofiltering) You can filter fields in the [row](#), [column](#), and [detail areas](#). When you filter a field, you select one or more [items](#) of data in the field that you want to view, and hide the other items.

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr4	\$2,000
Tennis	Qtr3	\$600
Tennis	Qtr4	\$1,500

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr4	\$2,000
Golf	Qtr3	\$6,430

Sport	Quarter	Sales
Golf	Qtr3	\$1,500
Golf	Qtr3	\$6,430

- 1 First, the Sport field is filtered to display only Golf sales ...
- 2 ... and then the Quarter field is filtered to display only Golf sales in Qtr3.
- 3 When you filter a field, the drop-down arrow ▼ for the filtered field changes to blue instead of black, and the **AutoFilter** button  on the toolbar is selected.

Filtering a field is particularly helpful when you have a large amount of source data but you want to focus on specific areas. When you filter a field, you can display the data for a single item, or you can select some items to display and other items to hide. You can filter on more than one field at a time to further narrow the focus.

Filtering based on the data in one cell (Filter by Selection) You can filter a field to display only data that matches the value in a selected cell. Filtering by selection is particularly useful for fields in the detail area, when you want to view all of the rows that contain a particular value.

Using a field in the Filter axis You can also filter data by using [filter fields](#). When you select one or more items in the filter field, the data that's displayed and calculated in the entire PivotTable view changes to reflect those items. You can add new filter fields or move existing fields to the filter area.



- 1 Filter field

Conditional filtering You can apply a filter to a row or column field to show the top or bottom n items based on a total. For example, you can filter for the three cities that generated the most sales or the five products that are least profitable. Instead of n items, you can also choose to filter for a certain percentage of items. For example, you can filter for the top or bottom 25% of salespeople based on the number of orders handled. If there are 40 salespeople, after the filter is applied, you will see data for 10 people.

[Combining multiple filters](#)

- Filters are additive. Each filter you apply is added to the ones you already have in effect. However, when you apply a [filter by selection](#), the [conditional filter](#) already applied on the field is removed before the filter by selection is applied.
- Conditional filters are independent of each other. The result of one conditional filter does not affect the result of another.
- An [autofilter](#) is always applied before a conditional filter, regardless of the order in which you applied them. For example, if you set a conditional filter to show the top two cities based on sales, followed by an autofilter on the ShippedCity field to include only five cities, the PivotTable view will show the top two of the five cities you selected.

[Removing and reapplying a filter](#)

You can display all data for all fields by turning off [autofiltering](#). However, the filter settings are retained, so that when you turn autofiltering back on, the data that was previously displayed or hidden is again displayed or hidden.

Filter settings are retained when you move fields to change the layout. Also, if you remove a field and later add the field back to the layout, the same items are again hidden.

If you don't want to retain your filter selections, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Identifying filters that are currently in effect](#)

- When items in a field are hidden by [autofiltering](#) or [filter by selection](#), the drop-down arrow  in the field label is blue. Click the blue arrow  to see the filter that is in effect.
- When items in a field are hidden by [conditional filtering](#), a funnel  appears to the left of the drop-down arrow . You can get more information about the current filter in the **Properties** dialog box.

[Effects of filtering on calculations](#)

Totals can include only the displayed data, or both visible and hidden data. The default is to exclude hidden data from totals. You can change the setting to include or exclude hidden data in totals. This setting affects only data that's hidden by filtering row and column fields; data that's hidden by using [filter fields](#) is always excluded from totals.

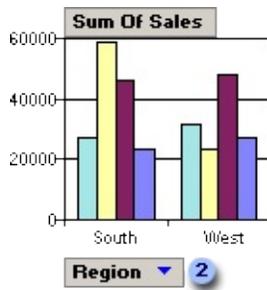
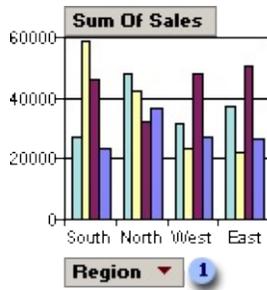
[About filtering data in a PivotChart view](#)

You can use the filter feature to find specific data values or all data that matches a value.

[Filtering techniques](#)

Filtering a field (Autofiltering) You can filter fields in the category, series, and filter areas. When you filter a field, you select one or more items of data in the field that you want to view, and hide the other items.

Filtering a field is particularly helpful when you have a large amount of source data but you want to focus on specific areas. When you filter a field, you can display the data for a single item, or you can select some items to display and other items to hide. You can filter on more than one field at a time to further narrow the focus.



1 Category field Region showing all items (not filtered)

2 Category field Region filtered to show South and West region items

Conditional filtering You can apply a filter to a [series](#) or [category field](#) to show the top or bottom n items based on a total. For example, you can filter for the three cities that generated the most sales or the five products that are least profitable. Instead of n items, you can also choose to filter for a certain percentage of items. For example, you can filter for the top 25% or bottom 25% of salespeople based on the number of orders handled. If there are 40 salespeople, after the filter is applied, you will see data for 10 people.

[Combining multiple filters](#)

Filters are additive. Each filter you apply is added to the ones you already have in effect.

[Removing and reapplying a filter](#)

You can display all data for all fields by turning off [autofiltering](#). However, the filter settings are retained, so that when you turn autofiltering back on, the same data that was previously displayed or hidden is again displayed or hidden.

Filter settings are retained when you move fields to change the layout. Also, if

you remove a field and later add the field back to the layout, the same items are again hidden.

If you don't want to retain your filter selections, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Identifying the filters that are currently in effect](#)

When items in a field are hidden by [autofiltering](#), the drop-down arrow  in the field label is blue. Click the blue arrow  to see the filter that is in effect.

[Show All](#)

About sorting records (ADP)

Some of the content in this topic may not be applicable to some languages.

The type of sort that you create depends on whether you want to sort records at the server before they are retrieved from the database, or locally after they are retrieved. It also depends on the type of object for which you're sorting.



[Sorting records before they are retrieved from the database](#)

You can sort records at the server by using a [view](#), [stored procedure](#), [user-defined function](#), or [SQL statement](#) in the **RecordSource** property of a form or report. You sort the records by using the **Sort Type** grid column or by including an ORDER BY clause in the query's SELECT statement. When you specify a sort order this way, you can perform complex sorts, which means you can sort records in ascending order by some fields and in descending order by others. Each time you view the results of a stored procedure, user-defined function, or SQL statement, the sort order is automatically applied. To sort data when the record source is a table, you must use a local sorting technique.



[Sorting records after they are retrieved from the database](#)

You can sort records locally in [Form view](#) of a form, in [Datasheet view](#) of a table, [query](#), or form, in [Page view](#) of a [data access page](#), in PivotChart view, and in PivotTable view. You do this by using the **Sort** buttons in a form, datasheet, Page, PivotChart, or PivotTable. You can also sort records locally by setting the **OrderBy** property in form [Design view](#), or by setting the **DefaultSort** property in Page Design view. For reports, you can sort groups of records locally by using the **Sorting and Grouping** box, or you can sort on all records locally by setting the report's **OrderBy** property in Design view.

When you use the **Sort** buttons, you can perform simple sorts, which means you can sort all records in ascending or descending order (but you can't use a different sort order on more than one field). When you set the **OrderBy** or **DefaultSort** property directly, you can perform complex sorts which means you can sort records in ascending order by some fields and in descending order by

others. Microsoft Access saves the local sort order that you specify in a form, report, PivotTable, or PivotChart and applies it automatically when you open the object or view. The local sort order for a table, query, or Page is not saved; however, if you create a form or report from the sorted table or query while it's open, the new form or report will inherit the sort order.

[Deciding whether to sort before or after database retrieval](#)

The choice of when to sort records can affect performance and even the results of your sort operation. Generally speaking, you should sort records before you retrieve them from the database, because the database is often on a server that is faster than your computer and because SQL Server is designed to sort efficiently. You might want to sort records after retrieving them from the database for ad hoc analysis and when the number of records is relatively small.

[Sorting results may vary](#)

Be aware that the results of sorting before you retrieve records from the database might differ from the results of sorting after you retrieve records from the database. This is because the sort order after records are retrieved from the database is determined by the regional settings in Windows **Control Panel** of your computer, whereas the sort order before records are retrieved from the database is determined by the settings on the computer (defined during SQL Server or SQL Server 2000 Desktop Engine installation) where the database is located.

[About sorting data in PivotTable or PivotChart view](#)

[In PivotTable view](#)

[Sort in ascending or descending order](#)

For example, you can sort a sales column in ascending order or an employee name column in descending order.

You can also sort data separately for each column. The order in which you sort the columns determines how the data will be organized.

Salesperson ▾	Product ▾	▾
Buchanan	Beverages	▬
Buchanan	Dairy	▬
Buchanan	Meat	▬
Buchanan	Produce	▬
Davolio	Beverages	▬
Davolio	Meat	▬
Davolio	Produce	▬
Suyama	Beverages	▬
Suyama	Dairy	▬
Suyama	Meat	▬
Suyama	Produce	▬

In this example, data is organized by salesperson. The Product column was sorted first so that the products would be in alphabetical order; the Salesperson column was sorted second.

When sorting multiple columns, first determine the order in which you want to sort the columns, and then work backward, sorting the outermost column last. For example, if you're sorting two columns, determine which is the outer column and which is the inner column. Sort the inner column first, and then sort the outer column. To list the products in alphabetical order for each salesperson, you would first sort the inner Product column and then sort the outer Salesperson column.

[Define a custom sort order](#)

You can define a custom order for a [row](#) or [column field](#) when sorting in ascending or descending order does not meet your requirements. For example, you might want to show the values in the Title column ordered on the basis of the seniority of the title. If the column had three values— Vice President, General Manager, and Manager— sorting in ascending or descending order will not show the data in the order you want. In this case, you might want to explicitly define the order of the items for the Title column.

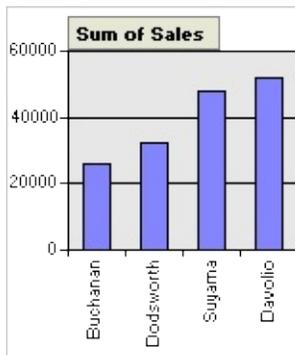
If items are added to a field after you define a custom sort order, the new items will appear at the bottom of the field until you rearrange them.

[In PivotChart view](#)

In a PivotChart view, you can sort numerical data or alphabetical data in ascending or descending order. For example, you can sort a [series](#) that contains numerical sales data to show the highest to lowest sales amounts, or you can sort a [category field](#) that contains company names to display in ascending order (A -

Z).

In the following example, the [series field](#), which consists of sales amounts, is sorted in ascending order to show lowest to highest sales. If the category field had been sorted in ascending order, the salespeople's names would be displayed on the category axis in alphabetical order instead.



If you have a stacked bar, area, or column chart, you can sort numerically based on the height of the entire stacked value rather than just one series.

[Show All](#)

Apply a filter (ADP)

[Apply a filter in a table, query, or form](#)

Do one of the following:

- To apply a filter in [Form view](#), [Datasheet view](#), or the [Filter By Form](#) window, click **Apply Filter**  on the toolbar.
- To apply a filter from the [Server Filter By Form](#) window, click **Apply Server Filter** .

Notes

- Microsoft Access can apply a filter automatically when you open a form if you set the form's **OnOpen** property to a macro that uses the `ApplyFilter` action or to an event procedure that uses the `ApplyFilter` method of the `DoCmd` object, or by setting the form's **ServerFilter** property without enabling **Server Filter By Form**.
- If you create a local filter in a form or subform that already contained a filter from a previous working session, the new filter replaces the previous one.

[Reapply a filter to show all updated records](#)

If you're working in a [multiuser \(shared\) database](#), you may want to reapply the filter occasionally to see which records have been added, deleted, or changed since you opened the form or datasheet and applied the filter.

[Reapply a filter that you created using Filter By Selection, Filter By Form, or Filter For Input](#)

- With the filtered records displayed in a form, subform, or datasheet, click **Apply Filter/Sort** on the **Records** menu.

[Reapply a filter that you created using Server Filter By Form](#)

- Click **Server Filter By Form**  on the toolbar to redisplay the window, and then click **Apply Server Filter**  on the toolbar.

[Reapply an existing filter to a data access page in Page view](#)

You can reapply a filter that you created during the current working session.

1. Open a data access page in [Page view](#).
2. Do one of the following:
 - On an ungrouped page, click **Filter Toggle**  on the record navigation toolbar so that it is selected.
 - On a [grouped page](#), expand the group that you want to filter, and then click **Filter Toggle** on the record navigation toolbar for that group so that the button is selected.

Note This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Show All](#)

Create a filter (ADP)

With a filter, you can limit a group of records to a specific subset that you want to view or work with. You can filter records at the server before they are retrieved from the database, or on your local computer after they are retrieved. You can also filter records in a [data access page](#) before or after database retrieval.

Limit records before they are retrieved from the database

[Enable or disable filters by using Server Filter By Form to limit records before they are retrieved for a form](#)

Note You can't use [Server Filter By Form](#) with a form bound to a [stored procedure](#).

1. Open a form in [Design view](#).
2. Display the property sheet for the form.
3. Do one of the following:
 - To enable Server Filter By Form, set the **ServerFilterByForm** property and **AllowFilters** property to **Yes**.
 - To disable Server Filter By Form, set the **ServerFilterByForm** property to **No**.
4. Switch to [Form view](#) or the form's [Datasheet view](#) to test using Server Filter By Form.

Note When you enable Server Filter By Form, Microsoft Access disables [Filter By Form](#).

[Use Server Filter By Form to retrieve a subset of records for a form](#)

When you are using [Server Filter By Form](#), no data is retrieved for the form until you specify the values that you want the filtered records to contain. You can specify [criteria](#) for the form or subform that's displayed. Each subform has its own **Look For** and **Or** tabs.

1. In the Server Filter By Form window, click the field in which you want to specify the criteria that records must meet to be included in the filtered set of records.
2. Enter your criteria by selecting the value that you're searching for from the list in the field (if the list includes field values), or by typing the value into the field.

[Find records in which a check box, toggle button, or option button is](#)

[or is not selected](#)

- Click the check box or button until it's the way you want it. To return it to a neutral position so that it won't be used as criteria for filtering records, continue clicking the check box or button until it's dimmed or unavailable.

[Find records in which a particular field is empty or not empty](#)

- Type **Is Null** or **Is Not Null** into the field.

Note You can select these options from the list in fields with a [Text](#) or [Image data type](#), and in [calculated fields](#) in queries. If the list in all fields displays **Is Null** and **Is Not Null** instead of values from the underlying table, the option to display lists in the Server Filter By Form window is cleared in the **Options** dialog box. You can change the Server Filter By Form settings to display field values in the list.

[Find records by using a criteria expression](#)

- Type the [expression](#) into the appropriate field using SQL Server expression syntax.

If you specify values in more than one field, the filter returns records only if they contain the same values that you specified in each of those fields.

3. To specify alternative values that records can have, and include those values in the filter's results, click the **Or** tab for the form or subform you're filtering, and enter more criteria.

The filter returns records if they have all the values that are specified on the **Look For** tab or all the values specified on the first **Or** tab, or all the values specified on the second **Or** tab, and so on.

4. Click **Apply Server Filter**  on the toolbar.

Notes

- Each time you apply the filter, Access automatically sets the form's

ServerFilter property to a string expression that represents the filter that you defined in the Server Filter By Form window.

- If you created a filter on a subform, that filter is also available when you open the form for the subform independently.
- If you're using Server Filter By Form, you can't use [Filter By Form](#).

[Create a specific server filter that's automatically applied when a form or report is opened](#)

You can't use a server filter with a form bound to a [stored procedure](#).

1. Open a form in [Design view](#).
2. Display the property sheet for the form
3. Set the **ServerFilter** property to a [string expression](#) consisting of a [WHERE clause](#) without the WHERE keyword. For example, to find all the records where the Country field contains the value "USA," enter **Country = 'USA'**.

[Create a filter to limit records in a data access page before they are retrieved from the database](#)

Filtering records before they are retrieved from the database can improve data access page performance because there are fewer records to download from the underlying database.

- You can base your page on a [query](#) that includes criteria (search conditions) to limit the records.
- In a [Microsoft Access database](#) you can define a server-side filter in the **ServerFilter** property of the **RecordsetDef** object, or by using the **Filter** property in Visual Basic for Applications (VBA) or Microsoft Visual Basic Scripting Edition (VBScript). In a [Microsoft Access project](#) you can define a server-side filter in the **ServerFilter** property in VBA or VBScript.
- On a grouped data access page, you can create a filter that retrieves data for a specific group, such as orders for a specific customer. The data access page won't display any records until you choose the group from a drop-down list in [Page view](#) or Microsoft Internet Explorer. Once you choose the group, the filter is applied and only those records are retrieved.

On an ungrouped data access page, you can create a filter by designating a field as the filter field. When you choose a value from the filter field, all records that contain the selected value are retrieved.

- In [Design view](#), you can add a server-side filter to a hyperlink on a page. Then, when you click the link in Page view or in the browser, only the records specified in the filter are displayed on the page you jump to.

Limit records after they are retrieved from the database

[Filter by selecting values in a form or datasheet after records are retrieved \(Filter By Selection\)](#)

1. In a field on a form, subform, or datasheet, find one instance of the value that you want records to contain in order to be included in the filter's results.
2. Select all or part of a value in a field by doing one of the following:

Note How you select the value determines what records the filter returns.

[Find records in which the entire contents of that field match the selection](#)

- Select the entire contents of a field or place the insertion point in a field without selecting anything.

For example, select the value "Berlin" in the City field to return all records with Berlin as the city.

[Find records in which the value in that field starts with the same characters you selected.](#)

- Select part of a value starting with the first character in a field.

For example, select "Fran" in the CompanyName field with the value "France restauration" to return all records that have a company name starting with "Fran," such as "Franchi S.p.A." and "Frankenversand".

[Find records in which all or any part of the value in that field contains the same characters you selected.](#)

- Select part of a value starting after the first character in a field.

For example, select the letters "Del " in the CompanyName field with

the value "Old World Delicatessen" to return all records that have "del" anywhere in the CompanyName field, such as "Ernst Handel", "Galería del gastrónomo", and "Que Delícia".

Note You can't find records based on the end of the value in a field; however, if you select a word or part of a word that doesn't represent the last character of that field, records that contain that word or part of a word will be returned. For example, say the Pub_Name field contains the values "Five Lakes Publishing" and "Publishing Times." If you select "Publishing," then of the two records only "Publishing Times" will be returned. However, if you select "Pub" or "Publish," the filter will return both records.

3. Select the value, and then click **Filter By Selection**  on the toolbar.
4. If you want to refine your search, you will need to reset the filter to show all of the records and then repeat steps 2 and 3. To reset the filter, click **Remove Filter**  on the toolbar.

Notes

- You can also filter for records that do not have a certain value. After selecting a value, right-click it, and then click **Filter Excluding Selection**.
- If you created a filter on a subform, that filter is also available when you open the form for the subform independently.

[Create a filter in Form view or Datasheet view \(Filter By Form\)](#)

1. Open a form in [Form view](#), or a table, [query](#), or form in [Datasheet view](#).
2. Click **Filter By Form**  on the toolbar to switch to the [Filter By Form](#) window. (If [Server Filter By Form](#) is enabled, Filter By Form will be disabled.)
3. You can specify [criteria](#) for the form, subform, or datasheet that's displayed. Each subform has its own **Look For** and **Or** tabs.
4. Click the field in which you want to specify the criteria that records must meet to be included in the filtered set of records.
5. Enter your criteria by selecting the value that you're searching for from the list in the field (if the list includes field values), or by typing the value into the field.

[Find records in which a check box, toggle button, or option button is or is not selected](#)

- Click the check box or button until it's the way you want it. To return it to a neutral position so that it won't be used as criteria for filtering records, continue clicking the check box or button until it's dimmed or unavailable.

[Find records in which a particular field is empty or not empty](#)

- Type **Is Null** or **Is Not Null** into the field.

Note You can select these options from the list in fields with a [Text](#) or [Image data type](#), and in [calculated fields](#) in queries. If the list in all fields displays **Is Null** and **Is Not Null** instead of values from the underlying table, the option to display values in the Filter By Form window is cleared in the **Options** dialog box. You can change the Filter By Form settings to display field values in the list.

[Find records by using a criteria expression](#)

- Type the [expression](#) into the appropriate field.

If you specify values in more than one field, the filter returns records only if they contain the same values that you specified in each of those fields.

6. To specify alternative values that records can have, and include those values in the filter's results, click the **Or** tab for the form, subform, or datasheet that you're filtering, and enter more criteria.

The filter returns records if they have all the values that are specified on the **Look For** tab, or all the values specified on the first **Or** tab, or all the values specified on the second **Or** tab, and so on.

7. Click **Apply Filter**  on the toolbar.

[Create a filter in Form view or Datasheet view by entering criteria in place \(Filter For Input\)](#)

1. Open a form in [Form view](#), or a table, query, or form in [Datasheet view](#).
2. Do one of the following:

[Specify the exact field value that you want the filtered records to contain](#)

- Right-click in the field in the form, subform, or datasheet you're filtering, and then in the **Filter For** box on the shortcut menu, type the value.

For example, to find all the records that have "London" in the City field, right-click in the City field, and then type **London** in the **Filter For** box.

[Specify more complex criteria](#)

- Type the complete [expression](#).
3. Do one of the following:
 - To apply the filter and close the shortcut menu, press ENTER.
 - To apply the filter and keep the shortcut menu displayed (so that you can specify additional [criteria](#) for the field), press TAB. Enter new criteria, and then press TAB again. Repeat until you have just the records you want.

[Enable or disable filtering in a form](#)

1. Display the property sheet for the form.
2. Set the **AllowFilters** property to **Yes** to enable filtering or to **No** to disable filtering capabilities in [Form view](#) and in form [Datasheet view](#).

Note When you set the **AllowFilters** property to **No**, you disable Filter By Selection, [Filter By Form](#), [Server Filter By Form](#), and Filter For Input.

[Create a filter in a data access page to limit records after they are retrieved from the database](#)

[Create a filter in Page view to limit records after they've been retrieved](#)

1. Open a data access page in [Page view](#).
2. Do one of the following:

[In an ungrouped page](#)

1. Move focus to the field that has the value that you want to use to filter records.
2. Click **Filter By Selection**  on the record navigation toolbar.

[In a grouped page](#)

1. Expand the group that you want to filter.
2. Move focus to the field that has the value that you want to use to filter records.
3. Click **Filter By Selection**  on the record navigation toolbar for that group.

The filter applies only to the current group. In other words, if a page groups customer orders by country/region, and you filter the records for Brazil that have a required date of 6/2/98, that filter applies only to the records for Brazil, not to the records for the other countries/regions.

3. To further narrow the subset of records, continue selecting fields and clicking **Filter By Selection**  until you have just the records you want.

For example, if you want to see records only for customers in Germany with a required date of 5/12/98, select "Germany" in the Country field, click **Filter By Selection**, select "5/12/98" in the Required Date field, and then click **Filter By Selection** again.

Notes

- If you select only part of the value in a field, the data access page still uses the entire field value for filtering.
- You can remove the filter and then reapply it during the same working session.

- This procedure doesn't apply to a [PivotTable list](#), [Spreadsheet Component](#), or [Chart Component](#). These features can be filtered independently from the rest of the data on the data access page and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Create a hyperlink that filters the records on a data access page that the link jumps to](#)

1. Open a data access page in [Design view](#).
2. On the **Insert** menu, click **Hyperlink**.
3. In the **Insert Hyperlink** dialog box, click **Page in this database** under **Link to**.
4. In the **Text to display** box, type the text that you want to be displayed in the field or text box. If you leave the box blank, Microsoft Access will use the hyperlink address as the display text.
5. Click **ScreenTip** and type the text that you want to appear when the user rests the pointer on the hyperlink. If you leave the **ScreenTip text** box blank, Access will display the hyperlink address as the ScreenTip.
6. From the list, select the data access page you want to link to.
7. To create a filter for the data access page, click **Server Filter** and type an [expression](#) in the form of a [WHERE clause](#) in the **Filter criteria for the data access page** box. For example, if you have a data access page that displays order information, including the ID of the salesperson, and you want to link to a data access page with the employee record about that salesperson, type **EmployeeID=[EmployeeID]**.
8. Click **OK** twice.

Access adds the hyperlink to the data access page. To test the link, switch to [Page view](#) and click the hyperlink. Access opens the target page in your Web browser.

[Filter data in a PivotTable or PivotChart view](#)

[Filter data in a PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Filter a field in the row, column, filter, or detail area \(Autofiltering\)](#)

1. Click the arrow  for the field that contains the [items](#) of data that you want to filter.
2. Select the check boxes for the items you want to display, and clear the check boxes for the items you want to hide.

Notes

- To display or hide all of the data for a field, select or clear the **(All)** check box. A shaded check box indicates that some of the lower-level items are hidden. To see which lower-level items are hidden, click the expand indicator  beside the item with the shaded check box.
- When placed in the [row](#), [column](#), or [detail area](#), a field always supports multiple selection. A field in the [filter area](#) can support either single or multiple selection. To switch from multiple to single selection, on the **Filter and Group** tab of the **Properties** dialog box, clear the **Allow selecting multiple items when in filter area** check box.

[Display only data that matches a cell \(Filter By Selection\)](#)

- Right-click the cell that contains the value you want to display, and then click **Filter By Selection** on the shortcut menu.

[Filter a row or column field for top or bottom *n* values or a percentage \(Conditional filtering\)](#)

1. Select the field that has the values you want to filter.
2. Click **Properties**  on the **PivotTable** toolbar, and then click the **Filter and Group** tab.
3. Set **Display the** to **Top** or **Bottom**.
4. Set **Items** to a number in the list, or type the value you want.
5. To show *n* number of items, clear the **%** check box. To show *n*% of items,

select the % check box.

6. Set **Based on** to a value in the list.

Notes

- When items in a field are hidden by an [autofilter](#) or a [filter by selection](#), the arrow  in the field label is blue. When items are hidden by a conditional filter, the filtered field displays a funnel icon  to the left of the arrow.
- When you apply a filter, the displayed totals are based on visible items only. You can change the setting to include hidden data.

 [Filter data in a PivotChart view](#)

 [Filter a field in the category, series, or filter area \(Autofiltering\)](#)

If the [filter area](#) does not have a field, you can either add a new field or move a field from another area to the filter area.

1. Open a datasheet or form in [PivotChart view](#).
2. Click the arrow  for the field that contains the items of data that you want to filter.
3. Select the check boxes for the items you want to display, and clear the check boxes for the items you want to hide.

Notes

- To display or hide all of the data for a field, select or clear the **(All)** check box. A shaded check box indicates that some of the lower-level items are hidden. To see which lower-level items are hidden, click the expand indicator  beside the item with the shaded check box.
- When placed in the category, series, or filter area, a field always supports multiple selection. A field in the filter area can support either single or multiple selection.

 [Filter a row or column field for top or bottom *n* values or a percentage \(Conditional filtering\)](#)

1. Open a datasheet or form in [PivotChart view](#).

2. Select the field, [series](#), or [point](#) that has the values you want to filter.
3. On the **PivotChart** toolbar, click **Properties** , and then click the **Filter and Group** tab.
4. In the **Display the** box, click **Top** or **Bottom**.
5. Set **Items** to a number in the list, or type the value you want.
6. To show n number of items, clear the % check box. To show $n\%$ of items, select the % check box.
7. Set **Based on** to a value in the list.

Note When items in a field are hidden by an [autofilter](#), the arrow  in the field label is blue .

[Show All](#)

Delete a local sort (ADP)

- In [Form view](#) or [Datasheet view](#), click **Remove Filter/Sort** on the **Records** menu.

[Show All](#)

Delete or remove a filter (ADP)

Deleting a filter permanently eliminates the filter. Removing a filter does not delete the filter. You remove a filter to show all records before you apply the filter. After removing a filter, you can apply it again.

[Delete a filter in the Server Filter By Form window and show all records](#)

You can only delete a filter created using [Server Filter By Form](#), you can't temporarily remove it and reapply the same one later on.

1. In the Server Filter By Form window, click **Clear Grid**  on the toolbar.
2. Click **Apply Server Filter**  on the toolbar.

[Remove a filter from a table, query, or form and show all records](#)

- To remove a filter and show the records that were displayed previously in the table, query, or form, click **Remove Filter**  on the toolbar in [Form view](#) or [Datasheet view](#).
- To remove a filter in a [multiuser \(shared\) database](#) and see which records have been added, deleted, or changed since you opened the table, query, or form, click **Remove Filter/Sort** on the **Records** menu in Form view or Datasheet view.

[Remove a filter to show all records in a data access page in Page view](#)

1. Open a data access page in [Page view](#).
2. Click **Filter Toggle**  on the record navigation toolbar so that it is not selected.

Notes

- Removing a [filter](#) doesn't delete it. You can reapply a filter during the current working session by clicking the **Filter Toggle** button on the record navigation toolbar so that it is selected.
- This procedure doesn't apply to a Microsoft Office PivotTable Component,

Spreadsheet Component, or Chart Component. These features can be filtered independently from the rest of the data on the data access page, and may also support additional filtering capabilities. For more information, click **Help**  on that component's toolbar.

[Remove a filter data in a PivotTable or PivotChart view](#)

[Remove a filter from a PivotTable view](#)

When items in a field are hidden by filtering a field or filtering by selection, the arrow  in the field label is blue. When items are hidden by filtering for top or bottom values, the filtered field displays a funnel icon  to the left of the arrow.

1. Open a datasheet or form in [PivotTable view](#).
2. Do one or more of the following:

[Remove all filters](#)

- Click the **AutoFilter** button  on the **PivotTable** toolbar so that it is not selected.

Note If you click **AutoFilter**  again to turn filtering back on, the previously set filters will be reapplied.

[Remove an autofilter, or a filter by selection, from a field](#)

- Click the blue arrow  of the field from which you want to remove the filter.
- Select the **(All)** check box and click **OK**.

[Remove a conditional filter from a field](#)

- Select the field from which you want to remove [conditional filtering](#).
- On the **PivotTable** toolbar, click **Show Top/Bottom Items** , and then click **Show All**.

Note If you don't want to retain your filters, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Remove a filter from a PivotChart view](#)

When items in a field are hidden by filtering a field or filtering by selection, the arrow  in the field label is blue.

1. Open a datasheet or form in [PivotChart view](#).
2. Do one of the following:

[Remove all filters](#)

- Click the **AutoFilter** button  on the **PivotChart** toolbar so that it is not selected.

Note If you click **AutoFilter**  again to turn filtering back on, the previously set filters will be reapplied.

[Remove an autofilter from a field](#)

1. Click the blue arrow  of the field from which you want to remove the filter.
2. Select the **(All)** check box and click **OK**.

[Remove a conditional filter from a field](#)

1. Select the field from which you want to remove [conditional filtering](#).
2. On the **PivotChart** toolbar, click **Show Top/Bottom Items** , and then click **Show All**.

Note If you don't want to retain your filters, make sure the **AutoFilter** button  is not selected before you start selecting items to filter. If the button is not selected, selecting new items to filter automatically turns filtering on and removes your former filter settings.

[Show All](#)

Modify a filter (ADP)

If a filter isn't giving you the results you need, you can modify it.

[Modify a filter by using Filter By Selection or Filter Excluding Selection](#)

You can modify the filter by adding more [criteria](#) or changing the existing criteria.

1. To further limit a set of records that already has a filter applied, start with step 2. To change the criteria previously defined for a filter, first remove the filter if it is currently applied.
2. In [Form view](#) or [Datasheet view](#), select a value in the form, subform, datasheet, or [subdatasheet](#) that you want to use as criteria, and then either click **Filter By Selection**  on the toolbar or right-click the value and click **Filter Excluding Selection** on the shortcut menu. Repeat this process until you have the records you want.

[Modify a filter by using Filter For Input](#)

You can modify the filter by adding more [criteria](#) or changing the existing criteria.

1. To further limit a set of records that already has a filter applied, start with step 2. To change the criteria previously defined for a filter, first remove the filter if it is currently applied.
2. In [Form view](#) or [Datasheet view](#), do one of the following:
 - Right-click the field you want to specify criteria for in the main form, datasheet, subform, or [subdatasheet](#), and then in the **Filter For** box on the shortcut menu, type the value you want to use as criteria.
 - Right-click any field in the main form, datasheet, subform, or subdatasheet, and specify the expression you want to use as criteria in the **Filter For** box.
3. Press TAB and repeat step 2 until you have the records you want.

[Modify a filter by using Filter By Form or Server Filter By Form](#)

You can modify the filter by adding to blank fields other values you want the filtered records to contain, deleting or changing values in a field, or making any of the following modifications.

[Clear all the fields](#)

- Click **Clear Grid**  on the toolbar.

[Delete an Or tab](#)

1. Click the **Or** tab you want to delete.
2. On the **Edit** menu, click **Delete Tab**.

[Show All](#)

Sort records (ADP)

You can sort records at the server before they are retrieved from the database or on your local computer after they are retrieved. The way you sort records depends on the object you're working with and how you sort it.

Note You cannot sort by the contents of a [text](#), [ntext](#), or [image](#) column.

[Sort records before they are retrieved from the database](#)

[Sort records in a view, stored procedure, or in-line function](#)

1. In the [Database window](#), click **Queries**  under **Objects**, click the view, [stored procedure](#), or in-line function you want to open, and then click **Design** on the Database window toolbar.
2. If you have not already, add to the Grid pane the columns or expressions that you want to sort by.

If you don't want the columns or expressions to be part of the query results, remove them as output columns.

3. In the Grid pane, locate the row containing the first data column or expression to sort by, and then in the **Sort Type** grid column, choose **Ascending** or **Descending**.

	Column	Table	Output	Sort Type	Sort Order
	emp_id	employees	<input checked="" type="checkbox"/>		
	fname	employees	<input checked="" type="checkbox"/>		
	minit	employees	<input checked="" type="checkbox"/>		
	lname	employees	<input checked="" type="checkbox"/>		
	hire_date	employees	<input checked="" type="checkbox"/>	Ascending	1
			<input type="checkbox"/>		

4. If you want to sort by multiple columns or expressions, specify the sort order in the **Sort Order** column of the grid.

For example, if you query an employee table, you can order the results by department (sort order = 1) and within each department by last name (sort

order = 2).

Note For a view and inline [user-defined function](#), Microsoft Access automatically adds a TOP 100 PERCENT clause to the [SQL statement](#).

[Sort records in a table function](#)

- In the [Database window](#), click **Queries**  under **Objects**, click the table function you want to open, and then click **Design** on the Database window toolbar.

In the SQL text editor, enter a SELECT statement with an ORDER BY clause. For example:

```
SELECT LastName, FirstName, Title FROM Employees ORDER BY  
Title, LastName
```

In this example, you order the results by title (sort order = 1) and within each title by last name (sort order = 2).

[Sort records by using an SQL statement in the RecordSource property of a form or report](#)

1. Open the form or report in [Design view](#).
2. Open the property sheet.

[How?](#)

1. Open a table, query, form, report, or [data access page](#) in [Design view](#).
2. Open the property sheet by doing one of the following:
 - On a form or report, double-click the [form selector](#) or the [report selector](#).



Form or report selector

- On a data access page, click **Select Page** on the **Edit** menu.
- On a table or query, click **Properties**  on the toolbar.

- For a section on a form or report, double-click the [section selector](#).



1 Section selector

- For a section on a data access page, double-click the section bar.



1 Section bar

- To set properties of a query field, click the cell in the **Field** row, and then click **Properties**  on the toolbar.
 - To set properties of a query field list, click anywhere in the list, and then click **Properties**  on the toolbar.
 - To set properties for the body of a data access page, click above the first section, and then click **Properties**  on the toolbar.
 - To set properties for a [control](#), click the control, and then click **Properties**  on the toolbar.
3. In the property sheet, click the property you want to set, and then do one of the following:
- If an arrow appears in the property box, click it and then select a value from the list.
 - Type a setting or [expression](#) in the property box.
 - If a **Build** button  appears next to the property box, click it to display a [builder](#) or list of builders.

[Tips](#)

- To get Help on any property in the property sheet, click the property and then press F1.
- If you need more space to enter or edit a property setting, press SHIFT+F2 to open the **Zoom** box.

Note If the **AllowDesignChanges** property of a form is set to **All Views**, you can also set form properties in [Form view](#) and [Datasheet view](#).

3. Enter an SQL SELECT statement with an ORDER BY clause. For example:

```
SELECT LastName, FirstName, Title FROM Employees ORDER BY  
Title, LastName
```

In this example, you order the results by title (sort order = 1) and within each title by last name (sort order = 2).

[Tip](#)

To make it easier to type the SELECT statement, right-click the **RecordSource** property and click **Zoom**, or click the **Build** button next to the **RowSource** property to open the Query Builder.

[Sort records after they are retrieved from the database in Form or Datasheet view](#)

[Sort records locally in Form view or Datasheet view](#)

1. Click the field you want to use for sorting records. To sort records in a subform, click the field you want to sort.
2. Click **Sort Ascending**  or **Sort Descending** .

Notes

- In a form, you can sort on only one field at a time; in a datasheet, you can select two or more adjacent columns at the same time, and then sort them. Access sorts records starting with the leftmost selected column.
- When you insert new records or edit existing ones, the sort order is automatically removed. You can resort the records when you're done.

[Sort records in a report](#)

[Specify a sort order](#)

1. Open the report in [Design view](#).

2. Click **Sorting And Grouping**  on the toolbar to display the **Sorting And Grouping** box.
3. In the first row of the **Field/Expression** column, select a field name, or type an [expression](#).

The field or expression in the first row is the first sorting level (the largest set). The second row is the second sorting level, and so on.

Note When you fill in the **Field/Expression** column, Microsoft Access sets the **Sort Order** to **Ascending**.

Notes

- You can sort on up to 10 fields or expressions in a report.
- The Customer Labels report in the Northwind sample database sorts records by the values in three fields. To view this report, point to **Sample Databases** on the **Help** menu, click either **Northwind Sample Database** or **Northwind Sample Access Project**, and then open the Customer Labels report in Design view.

[Reapply an inherited sort order](#)

1. Open the report in [Design view](#), or if the report is already open, make sure it is [selected](#).
2. Click **Properties**  on the toolbar to display the report's property sheet.
3. In the **OrderByOn** property box, click **Yes** to reapply it.

Note If you specify a sort order in a report wizard or in the **Sorting And Grouping** dialog box in report Design view, it overrides the inherited sort order.

[Sort records on a data access page](#)

[Sort records on a Page](#)

You can sort data on a data access page, including data that's grouped. For grouped data access pages, you can sort the list of groups or the records within a group.

1. Open a data access page in [Page view](#).
2. Do one of the following:
 - In an ungrouped page, click the field you want to use for sorting records.
 - In a grouped page, expand the group that you want to sort, and then click the field you want to sort records on.

Note When you sort records within a group, the sort order applies only to that group. It does not affect the order of records in other groups.

3. Do one of the following:
 - To sort in ascending order, click **Sort Ascending** .
 - To sort in descending order, click **Sort Descending** .

[Tip](#)

If the records are displayed in tabular format, simply click the label of the field you want to sort records on. The records will display in ascending order. To sort records in descending order, click the label again.

Notes

- When you close the data access page, you'll lose the sort order. You can sort the records again when you open the page.
- The order in which numbers, text, and special characters are sorted depends on the database and your regional settings in Microsoft Windows Control Panel.
- This procedure doesn't apply to a Microsoft Office PivotTable Component, Spreadsheet Component, or Chart Component. These features can be sorted independently from the rest of the data on the data access page, and may also support additional sorting capabilities. For more information, click **Help**  on the toolbar for the PivotTable list, spreadsheet, or chart, and then search for "sorting."

[Change the default sort order on a Page](#)

1. Open the data access page in [Design view](#).
2. Scroll to a [section](#) that belongs to the group for which you want to specify

the sort order.

3. Right-click the [section bar](#), and then click **Group Level Properties**.
4. In the **DefaultSort** property box, type the name of the field you want to sort on. If you want to sort on multiple names, separate each field name with a comma. If you want to sort in ascending or descending order, follow each name with a space and the keyword ASC or DESC, respectively. For example, type **OrderDate DESC, OrderID ASC**. If you don't specify the order, Microsoft Access sorts in ascending order.

Note If the control you want to sort on is bound to an [expression](#), you must type the alias of the control as displayed in its **ControlSource** property. For example, if the **ControlSource** property of the control you want to sort on is set to **GroupOfExtendedPrice: ExtendedPrice**, type **GroupOfExtendedPrice** in the **DefaultSort** property box.

[Sort data in PivotTable or PivotChart view](#)

[Sort data in ascending or descending order](#)

You can sort data that is displayed in natural or [custom order](#) in ascending or descending order.

1. Open a datasheet or form in [PivotTable view](#) or [PivotChart view](#).
2. Do one of the following:

[In PivotTable view](#)

- To sort the data in a field based on the [item](#) labels, click the label of the field that contains the data you want to sort.
- To sort the data in a field based on the summary values in a column, click a cell in the column that contains the values that you want to sort on.
- To sort the data in a field based on subtotal or grand total values, click a subtotal or grand total value for the field that contains the data you want to sort.

Note If you click a grand total value in the bottom row, the columns will be

sorted instead of the rows.

[In PivotChart view](#)

- To sort numeric data in a series, click the [series](#) or a [series point](#) within the series you want to sort.
 - To sort items in a field alphabetically, click the [category field](#) or [series field](#), or click a category axis label.
3. On the **PivotTable** or **PivotChart** toolbar, click **Sort Ascending**  or **Sort Descending** .

Notes

- To sort on multiple fields, start by sorting the innermost field first and work backward, sorting the outermost field last.

[Specify a custom sort order in PivotTable view](#)

1. Open a datasheet or form in [PivotTable view](#).
2. Select the item you want to move.
3. Drag the selected item and drop it over the item above which you want to place it.
4. Repeat steps 2 and 3 until you have the items in the order you want.

Notes

- To switch from ascending or descending sort to custom sort, click **Sort Ascending**  or **Sort Descending**  on the **PivotTable** toolbar so that the button is not selected.
- If a field with custom sorting has a parent field, each parent member will display the lower-level members in custom order.

Note The order in which numbers, text, and special characters are sorted depends on the source data and your regional settings in Microsoft Windows Control Panel.

[Show All](#)

Troubleshoot filters (ADP)

[The filter returns too many records.](#)

[Filter By Form](#)

- Enter a value or [expression](#), or pick a value from the list in other fields on the **Look For** tab.
- Delete some of the alternative [criteria](#) that records can meet by deleting one or more **Or** tabs.

[Filter By Selection](#)

- Select another value that you want the records to contain, and then click **Filter By Selection**  on the toolbar. Repeat this until you have just the records you want to see.

[Filter Excluding Selection](#)

Make sure you haven't selected values in more than one row at a time. Although you can do this, Microsoft Access interprets the filter [criteria](#) as follows: "exclude the first selected value OR the second selected value," instead of "exclude the first selected value AND the second selected value."

Instead, select the first value you want to exclude, right-click, and then select **Filter Excluding Selection**. Select the next value you want to exclude, and then repeat the previous step.

[Filter For Input](#)

- Specify another value in the **Filter For** box, and then press ENTER. Repeat until you have the records you want.

[The filter returns too few records.](#)

[Filter By Form](#)

- Add alternative [criteria](#) that records can meet on the **Or** tab (at the bottom of the window).
- If you have multiple criteria on the **Look For** or **Or** tab, delete the value or expression that's restricting the records you want to see.

[Filter By Selection](#)

- Remove the filter, select a value you're looking for, and then click **Filter By Selection**  on the toolbar. If necessary, repeat this until you have the records you want to see.
- If you're using Filter By Selection on a combo box field that gets its values from a value list, check if the bound column for the combo box is the first column displayed in the list. If it's not and you've selected as your [criteria](#) part of a value from the combo box's displayed column, Microsoft Access can't match the criteria to the data in the bound column. You need to either change the combo box so the bound column is the first column displayed in the list, or fill the combo box with values from a table or query.

[Filter For Input](#)

Remove the filter if it's currently applied, and then specify a different value or expression in the **Filter For** box.

[The filter returns the wrong records.](#)

If the filter returns the wrong records, make sure you're using the correct fields to specify the value you're searching for or to specify other [criteria](#), and make sure you've entered the correct value or criteria. If you're filtering on fields from linked tables, the values you use in criteria for those fields are [case-sensitive](#) — they must match the case of the values in the underlying table.

If you're using expressions in the [Filter By Form](#) window verify you're using the correct [operators](#).

[I created a filter on a subform in which the records were synchronized with the main form, but the filter didn't have any affect on the main form.](#)

The filter behaved correctly. When you create a filter on a subform, it only applies to the subform. To filter the records on the main form, click the main form, and then create the filter.

[I created a filter on a subdatasheet, but the filter didn't have any effect on the datasheet that displays the subdatasheet.](#)

The filter behaved correctly. When you create a filter on a [subdatasheet](#), it only applies to the subdatasheet. To filter the records on the datasheet that displays the subdatasheet, click in that datasheet, and then create the filter.

[I can't use some of the fields in the Filter By Form or Server Filter By Form window.](#)

You can't use the following types of fields in the [Filter By Form](#) window or [Server Filter By Form](#) window because they don't apply to filtering records:

- A [calculated control](#) in a form.
- A field created with an [unbound control](#) or [unbound object frame](#).

[The local filter didn't return all the records that meet the criteria I specified.](#)

If you don't need to filter on all the records in the database, create a server filter to limit records to the subset that you need. If you do need to filter on all the records, raise the [maximum record limit](#). This limit determines how many records can be retrieved from the database. If the limit is too low, you aren't filtering on a complete set of records. For example, if you specify >M in the Last Name field to find all last names that start with M or letters after M in the alphabet, Microsoft Access may have only retrieved records containing last names that start with A through F.

[Show All](#)

Troubleshoot sort (ADP)

Some of the content in this topic may not be applicable to some languages.

[I applied a local sort order and now it looks like some records are missing.](#)

The [maximum number of records](#) determines how many records can be retrieved from the database. If the limit is too low, you aren't sorting on a complete set of records. If you need to work with all the records, raise the limit. If you don't need to work with all the records in the database, create a server filter to limit records to the subset you need.

[I receive replication errors after I change the language sort order.](#)

If you're synchronizing between replicas in a Microsoft SQL Server database or SQL Server 2000 Desktop Engine, you might encounter synchronization conflicts if the databases don't have the same language sort orders.

[Show All](#)

About upsizing a Microsoft Access database

Upsizing is the process of migrating some or all [database objects](#) from a [Microsoft Access database](#) (.mdb) to a new or existing Microsoft SQL Server database or new [Microsoft Access project](#) (.adp).

[Ways to upsize](#)

The Upsizing Wizard upsizes an Access database to a new or existing SQL Server 2000, 7.0, and 6.5 database or new Access project by migrating data and data definitions, and by moving database objects to the new database structure. There are three ways to use the Upsizing Wizard:

- Upsize all database objects from an Access database to an Access project so that you can create a client/server application. This approach requires some additional application changes and modification to code and complex queries.
- Upsize only data or data definitions from an Access database to an SQL Server database.
- Create an Access database front-end to an SQL Server database back-end so that you can create a [front-end/back-end application](#). This approach requires very little application modification since the code is still using the [Jet database engine](#).

[When to upsize an Access database to SQL Server](#)

Over time, most database applications grow, become more complex, and need to support more users. At some point in the life of your [Access database](#) application, consider upsizing to SQL Server to optimize database and application performance, scalability, availability, security, reliability, and recoverability.

[High performance and scalability](#)

In many situations, SQL Server offers better performance than an Access database. SQL Server also provides support for very large, terabyte-sized databases, which is much larger than the current limit for an Access database of two gigabytes. Finally, SQL Server works very efficiently on Microsoft Windows 2000 or later by processing queries in parallel (using multiple native threads within a single process to handle user requests) and minimizing additional memory requirements when more users are added.

[Increased availability](#)

SQL Server allows you to do a dynamic backup, either incremental or complete, of the database while it's in use. Consequently, you do not have to force users to exit the database to back up data. This means your database can be running up to 24 hours a day, seven days a week.

[Improved security](#)

Using a trusted connection, SQL Server can integrate with the Windows 2000 or later system security to provide a single access to the network and the database, employing the best of both security systems. This makes it much easier to administer complex security schemes. A SQL Server database on a server also employs innovative security features, which helps prevent unauthorized users from getting to the database file directly, but rather they must access the server first.

[Immediate recoverability](#)

In case of system failure (such as an operating system crash or power outage), SQL Server has an automatic recovery mechanism that recovers a database to the last state of consistency in a matter of minutes, with no database administrator intervention. Critical applications can be up and running again right away.

[Server-based processing](#)

Microsoft designed SQL Server from the beginning as a client/server database, where data and indexes reside on a single server computer that is often accessed over the network by many client computers. SQL Server reduces network traffic by processing database queries on the server before sending results to the client.

Thus, your client/server application can do processing where it's done best, on the server.

Your application can also use [user-defined functions](#), [stored procedures](#), and [triggers](#) to centralize and share application logic, business rules and policies, complex queries, and data validation and referential integrity code on the server, rather than on the client.

[Before you upsize an Access database](#)

Before you upsize your [Access database](#) to a SQL Server database or [Access project](#), consider doing the following:

- **Back up your database** Although the Upsizing Wizard doesn't remove any data or database objects from your Access database, it's a good idea to create a backup copy of your Access database before you upsize it.
- **Ensure you have adequate disk space** You must have adequate disk space on the device that will contain the upsized SQL Server database. The Upsizing Wizard works best when there is plenty of disk space available.

For a SQL Server 7.0 or later database, SQL Server automatically grows your database for you as it's being created.

For a SQL Server 6.5 database, multiply the size of your Access database by two and reserve that amount of space on your disk. This ensures that the Upsizing Wizard has enough space to upsize your database and also leave it some room to grow. If you expect to add a lot of data to the database, make the multiple larger.

- **Create unique indexes** A linked table must have a unique index to be updateable in Microsoft Access. The Upsizing Wizard can upsize an existing unique index, but can't create one where none exists. If you want to be able to update your tables, make sure you add a unique index to each Access table before upsizing.
- **Set a default printer** You must set a default printer to use the Upsizing Wizard if you want it to automatically create a [report snapshot](#) of the Upsizing Wizard report.

- **Assign yourself appropriate permissions on the Access database** You need READ and DESIGN permission on all database objects to upsize them.
- **Assign yourself appropriate permissions on the SQL Server database**
 - To upsize to an existing database, you need CREATE TABLE and CREATE DEFAULT permissions.
 - To build a new database, you need CREATE DATABASE permission, and SELECT permissions on the system tables in the Master database.
 - To create new devices, you must be a system administrator.
- **If desired, create multiple disk devices** If you are upsizing to a SQL Server version 6.5 database, you may want to create devices before running the Upsizing Wizard. The Upsizing Wizard creates all new devices on the same physical disk where the Master database device resides. If your server has multiple disks, you can place your database on one disk and the transaction log on a different disk. In the event of a disk failure, you can recover your database. Make sure that you have devices that are big enough on both disks. Create new devices if necessary.

SQL Server version 6.5 also allows databases and transaction logs to span several devices. However, the Upsizing Wizard only allows you to specify one device for your database and one device for the transaction log. To specify multiple devices for a database or transaction log, make those devices (and no other devices) the default devices. Then run the Upsizing Wizard, and select the default database or transaction log device.

Note If the size of the new SQL Server database or the size of the transaction log doesn't require using all the devices set as default, SQL Server uses only the devices necessary to accommodate the database or transaction log.

[The Upsizing Wizard report](#)

The Upsizing Wizard creates a report that provides a detailed description of all objects created, and reports any errors encountered during the process. The Upsizing Wizard automatically creates this report as a [report snapshot](#) with the same name as the Access database (for example, Northwind.snp), and stores it in

the default database folder.

Security This report may contain private, confidential, or sensitive information. Make sure this file is protected from being accessed by a malicious user.

The Upsizing Wizard report contains information about the following:

- Database details, including database size, as well as transaction logs and device names and sizes for a SQL Server 6.5 database.
- Upsizing parameters, including what table attributes you chose to upsize and how you upsized.
- Table information, including a comparison of Access and SQL Server values for names, data types, indexes, validation rules, defaults, triggers, and whether or not time stamps were added.
- Any errors encountered, such as database or transaction log full, inadequate permissions, device or database not created, table, default, or validation rule skipped, relationship not enforced, query skipped (because it cannot be translated to SQL Server syntax), and [control](#) and [record source](#) conversion errors in forms and reports.

[How database objects get upsized](#)

The following data and database objects get upsized:

[Data and data types](#)

For SQL Server version 7.0 or later databases, the Upsizing Wizard converts Access database text to Unicode by adding the Unicode string identifier to all string values and by adding the Unicode n prefix to all data types.

All Access database data types are converted to their equivalent in SQL Server.

[Queries](#)

For SQL Server 2000 databases:

- Select queries that don't have an ORDER BY clause or parameters are converted to views.
- Action queries are converted to stored procedure action queries. Access

adds SET NOCOUNT ON after the parameter declaration code to make sure the stored procedure runs.

- Select queries that only reference tables (also called base queries) that use either parameters or an ORDER BY clause are converted to user-defined functions. If necessary, the TOP 100 PERCENT clause is added to a query that contains an ORDER BY clause.
- Parameter queries that use named parameters maintain the original text name used in the Access database and are converted either to stored procedures or inline user-defined functions.

For SQL server 7.0 and 6.5 databases:

- Select queries are converted to views.
- Sorted queries are converted to a combination of views and stored procedures to allow for nesting and sorting (Views can be nested but can't contain ORDER BY clauses; stored procedures can contain ORDER BY clauses but can't be nested).
- Parameter queries, queries that depend on a parameter query, and action queries are converted to stored procedures.

Note For all SQL Server databases, you may need to manually convert queries that did not upsize, such as SQL Passthrough queries, data definition queries, and crosstab queries. You may also have to manually upsize queries that were nested too deeply.

 [Forms, reports, and controls](#)

For all SQL Server databases, SQL statements in **RecordSource**, **ControlsSource** and **RowSource** properties for forms, reports, or controls are kept in place and not converted to a stored procedure or user-defined function.

 [Data access pages](#)

The Upsizing Wizard changes the OLE DB connection and the data binding information in the Microsoft Office data source control to work with the new SQL Server database, and copies the page's corresponding HTML file to the same location as the Access project, renaming the HTML file with a "_CS"

suffix. The new pages in the Access project retain the original name, so that hyperlinks between the Access project Pages continue to work.

[Customized command bars and startup properties](#)

For all versions of SQL Server, the Upsizing Wizard upsizes any custom command bars as well as the following startup properties:

StartUpShowDBWindow
StartUpShowStatusBar
AllowShortcutMenus
AllowFullMenus
AllowBuiltInToolbars
AllowToolbarChanges
AllowSpecialKeys
UseAppIconForFrmRpt
AppIcon
AppTitle
StartUpForm
StartUpMenuBar
StartupShortcutMenuBar

[Modules and macros](#)

The Upsizing Wizard doesn't make any changes to modules and macro. Designing and building an optimized client/server application is different from designing and building an optimized file server application. Therefore, you will probably need to retrofit your application to take full advantage of SQL Server and an Access project. After you upsize, you need to manually convert code that uses recordsets from [Data Access Objects \(DAO\)](#) to [ActiveX Data Objects \(ADO\)](#) in your modules. You should also revise any table and query design code (The Upsizing Wizard does not upsize SQL Data Definition Language).

[Show All](#)

Upsize your Microsoft Access database

1. Open the Microsoft Access database you want to upsize.

[How?](#)

1. On the **File** menu, click **Open** .
2. Click a shortcut in the left side of the **Open** dialog box, or in the **Look in** box, click the drive or folder that contains the [Microsoft Access database](#) that you want.
3. In the folder list, double-click folders until you open the folder that contains the database.

If you can't find the database that you want to open, click **Tools** on the toolbar in the **Open** dialog box, and then click **Search** . In the **Search** dialog box, enter additional search criteria.

4. Do one of the following:
 - Double-click the database.
 - To open the database for shared access in a [multiuser](#) environment, so you and other users can read and write to the database, click **Open**.
 - To open the database for read-only access so that you can view it but cannot edit it, click the arrow next to the **Open** button, and then click **Open Read-Only**.
 - To open the database with [exclusive](#) access, click the arrow next to the **Open** button, and then click **Open Exclusive**.
 - To open the database for read-only access and also prevent other users from opening it, click the arrow next to the **Open** button, and then click **Open Exclusive Read-Only**.

Note You can directly open a data file in an external file format such as dBASE, Paradox, Microsoft Exchange, or Microsoft Excel. You can also directly open any [ODBC data source](#), such as Microsoft SQL Server or

Microsoft FoxPro. Access automatically creates a new Access database in the same folder as the data file and adds links to each table in the external database.

Tips

- To open one of the last several databases you had open, click the file name at the bottom of the **File** menu. Microsoft Access opens the database with the same option settings it had the last time you opened it. If the list of recently used files isn't displayed, click **Options** on the **Tools** menu, click the **General** tab, and then select the **Recently used file list** check box.
 - To view a list of shortcuts to databases that you have opened previously, click **History** on the left side of the **Open** dialog box.
 - To quickly locate a file or folder in the Favorites folder, click **Favorites** on the left side of the **Open** dialog box. To open a favorite file or folder, click the shortcut and then click **Open**.
2. On the **Tools** menu, point to **Database Utilities**, and then click **Upsizing Wizard**.
 3. Follow the directions in the Upsizing Wizard pages and dialog boxes, and then click **Finish** when you are done.
 4. If you need more information to decide what to do in each page of the wizard or dialog box, click **Help**.

The Upsizing Wizard will run to completion after you click **Finish**; there are no additional prompts. This is especially useful if you want to upsize a large database unattended. If you need to cancel the upsizing process, click **Cancel** on the status box that the Upsizing Wizard displays after you click **Finish**. Any error messages are written to the Upsizing Wizard report, created as a [report snapshot](#) with the same name as the Access database (for example, Northwind.snp), and stored it in the default database folder.

[Show All](#)

Use the Upsizing Wizard

[About upsizing a Microsoft Access database](#)

Upsizing is the process of migrating some or all [database objects](#) from a [Microsoft Access database](#) (.mdb) to a new or existing Microsoft SQL Server database or new [Microsoft Access project](#) (.adp).

[Ways to upsize](#)

The Upsizing Wizard upsizes an Access database to a new or existing SQL Server 2000, 7.0, and 6.5 database or new Access project by migrating data and data definitions, and by moving database objects to the new database structure. There are three ways to use the Upsizing Wizard:

- Upsize all database objects from an Access database to an Access project so that you can create a client/server application. This approach requires some additional application changes and modification to code and complex queries.
- Upsize only data or data definitions from an Access database to an SQL Server database.
- Create an Access database front-end to an SQL Server database back-end so that you can create a [front-end/back-end application](#). This approach requires very little application modification since the code is still using the [Jet database engine](#).

[When to upsize an Access database to SQL Server](#)

Over time, most database applications grow, become more complex, and need to support more users. At some point in the life of your [Access database](#) application, consider upsizing to SQL Server to optimize database and application performance, scalability, availability, security, reliability, and recoverability.

[High performance and scalability](#)

In many situations, SQL Server offers better performance than an Access database. SQL Server also provides support for very large, terabyte-sized databases, which is much larger than the current limit for an Access database of two gigabytes. Finally, SQL Server works very efficiently on Microsoft Windows 2000 by processing queries in parallel (using multiple native threads within a single process to handle user requests) and minimizing additional memory requirements when more users are added.

[Increased availability](#)

SQL Server allows you to do a dynamic backup, either incremental or complete, of the database while it's in use. Consequently, you do not have to force users to exit the database to back up data. This means your database can be running up to 24 hours a day, seven days a week.

[Improved security](#)

Using a trusted connection, SQL Server can integrate with the Windows 2000 system security to provide a single integrated access to the network and the database, employing the best of both security systems. This makes it much easier to administer complex security schemes. A SQL Server database on a server also employs innovative security features to help prevent unauthorized users from getting to the database file directly, but rather they must access the server first.

[Immediate recoverability](#)

In case of system failure (such as an operating system crash or power outage), SQL Server has an automatic recovery mechanism that recovers a database to the last state of consistency in a matter of minutes, with no database administrator intervention. Critical applications can be up and running again right away.

[Server-based processing](#)

Microsoft designed SQL Server from the beginning as a client/server database, where data and indexes reside on a single server computer that is often accessed over the network by many client computers. SQL Server reduces network traffic by processing database queries on the server before sending results to the client. Thus, your client/server application can do processing where it's done best, on

the server.

Your application can also use [user-defined functions](#), [stored procedures](#), and [triggers](#) to centralize and share application logic, business rules and policies, complex queries, and data validation and referential integrity code on the server, rather than on the client.

[Before you upsize an Access database](#)

Before you upsize your [Access database](#) to a SQL Server database or [Access project](#), consider doing the following:

- **Back up your database** Although the Upsizing Wizard doesn't remove any data or database objects from your Access database, it's a good idea to create a backup copy of your Access database before you upsize it.
- **Ensure you have adequate disk space** You must have adequate disk space on the device that will contain the upsized SQL Server database. The Upsizing Wizard works best when there is plenty of disk space available.

For a SQL Server 7.0 or later database, SQL Server automatically grows your database for you as it's being created.

For a SQL Server 6.5 database, multiply the size of your Access database by two and reserve that amount of space on your disk. This ensures that the Upsizing Wizard has enough space to upsize your database and also leave it some room to grow. If you expect to add a lot of data to the database, make the multiple larger.

- **Create unique indexes** A linked table must have a unique index to be updateable in Microsoft Access. The Upsizing Wizard can upsize an existing unique index, but can't create one where none exists. If you want to be able to update your tables, make sure you add a unique index to each Access table before upsizing.
- **Set a default printer** You must set a default printer to use the Upsizing Wizard if you want it to automatically create a [report snapshot](#) of the Upsizing Wizard report.
- **Assign yourself appropriate permissions on the Access database** You

need READ and DESIGN permission on all database objects to upsize them.

- **Assign yourself appropriate permissions on the SQL Server database**
 - To upsize to an existing database, you need CREATE TABLE and CREATE DEFAULT permissions.
 - To build a new database, you need CREATE DATABASE permission, and SELECT permissions on the system tables in the Master database.
 - To create new devices, you must be a system administrator.
- **If desired, create multiple disk devices** If you are upsizing to a SQL Server version 6.5 database, you may want to create devices before running the Upsizing Wizard. The Upsizing Wizard creates all new devices on the same physical disk where the Master database device resides. If your server has multiple disks, you can place your database on one disk and the transaction log on a different disk. In the event of a disk failure, you can recover your database. Make sure that you have devices that are big enough on both disks. Create new devices if necessary.

SQL Server version 6.5 also allows databases and transaction logs to span several devices. However, the Upsizing Wizard only allows you to specify one device for your database and one device for the transaction log. To specify multiple devices for a database or transaction log, make those devices (and no other devices) the default devices. Then run the Upsizing Wizard, and select the default database or transaction log device.

Note If the size of the new SQL Server database or the size of the transaction log doesn't require using all the devices set as default, SQL Server uses only the devices necessary to accommodate the database or transaction log.

[The Upsizing Wizard report](#)

The Upsizing Wizard creates a report that provides a detailed description of all objects created, and reports any errors encountered during the process. The Upsizing Wizard automatically creates this report as a [report snapshot](#) with the same name as the Access database (for example, Northwind.snp), and stores it in the default database folder.

The Upsizing Wizard report contains information about the following:

- Database details, including database size, as well as transaction logs and device names and sizes for a SQL Server 6.5 database.
- Upsizing parameters, including what table attributes you chose to upsize and how you upsized.
- Table information, including a comparison of Access and SQL Server values for names, data types, indexes, validation rules, defaults, triggers, and whether or not time stamps were added.
- Any errors encountered, such as database or transaction log full, inadequate permissions, device or database not created, table, default, or validation rule skipped, relationship not enforced, query skipped (because it cannot be translated to SQL Server syntax), and [control](#) and [record source](#) conversion errors in forms and reports.

[How database objects get upsized](#)

The following data and database objects get upsized:

[Data and data types](#)

For SQL Server version 7.0 or later databases, the Upsizing Wizard converts Access database text to Unicode by adding the Unicode string identifier to all string values and by adding the Unicode n prefix to all data types.

All Access database data types are converted to their equivalent in SQL Server.

[Queries](#)

For SQL Server 2000 databases:

- Select queries that don't have an ORDER BY clause or parameters are converted to views.
- Action queries are converted to stored procedure action queries. Access adds SET NOCOUNT ON after the parameter declaration code to make sure the stored procedure runs.
- Select queries that only reference tables (also called base queries) that use either parameters or an ORDER BY clause are converted to user-defined

functions. If necessary, the TOP 100 PERCENT clause is added to a query that contains an ORDER BY clause.

- Parameter queries that use named parameters maintain the original text name used in the Access database and are converted either to stored procedures or inline user-defined functions.

For SQL server 7.0 and 6.5 databases:

- Select queries are converted to views.
- Sorted queries are converted to a combination of views and stored procedures to allow for nesting and sorting (Views can be nested but can't contain ORDER BY clauses; stored procedures can contain ORDER BY clauses but can't be nested).
- Parameter queries, queries that depend on a parameter query, and action queries are converted to stored procedures.

Note For all SQL Server databases, you may need to manually convert queries that did not upsize, such as SQL Passthrough queries, data definition queries, and crosstab queries. You may also have to manually upsize queries that were nested too deeply.

[Forms, reports, and controls](#)

For all SQL Server databases, SQL statements in **RecordSource**, **ControlsSource** and **RowSource** properties for forms, reports, or controls are kept in place and not converted to a stored procedure or user-defined function.

[Data access pages](#)

The Upsizing Wizard changes the OLE DB connection and the data binding information in the Microsoft Office data source control to work with the new SQL Server database, and copies the page's corresponding HTML file to the same location as the Access project, renaming the HTML file with a "_CS" suffix. The new pages in the Access project retain the original name, so that hyperlinks between the Access project Pages continue to work.

[Customized command bars and startup properties](#)

For all versions of SQL Server, the Upsizing Wizard upsizes any custom command bars as well as the following startup properties:

StartUpShowDBWindow
StartUpShowStatusBar
AllowShortcutMenus
AllowFullMenus
AllowBuiltInToolbars
AllowToolbarChanges
AllowSpecialKeys
UseAppIconForFrmRpt
AppIcon
AppTitle
StartUpForm
StartUpMenuBar
StartupShortcutMenuBar

[Modules and macros](#)

The Upsizing Wizard doesn't make any changes to modules and macro. Designing and building an optimized client/server application is different from designing and building an optimized file server application. Therefore, you will probably need to retrofit your application to take full advantage of SQL Server and an Access project. After you upsize, you need to manually convert code that uses recordsets from [Data Access Objects \(DAO\)](#) to [ActiveX Data Objects \(ADO\)](#) in your modules. You should also revise any table and query design code (The Upsizing Wizard does not upsize an SQL Data Definition Language). For more information on developing optimized client/server applications, see the *Microsoft Office 2000/Visual Basic Programmer's Guide*.

[Choose to upsize to an existing database or new database](#)

[Use existing database](#)

If you select **Use existing database**, Microsoft Access displays the **ODBC Select Data Source** dialog box so you can create an [ODBC](#) connection to the existing Microsoft SQL Server database. For more information about ODBC data sources, Press **Help**.

[Create new database](#)

If you select **Create new database**, Microsoft Access displays the Microsoft SQL Server Database Wizard. Follow the instructions in the SQL Server Database Wizard.

What SQL Server would you like to use for this database? Type the name of the server you would like to use.

Use Trusted Connection You can use a trusted connection, that is, SQL Server can integrate with the Windows 2000 operating system security to provide a single log on to the network and the database.

Login ID and Password If you don't use a trusted connection, type the logon ID and password of an account with CREATE DATABASE privileges on the server.

What do you want to name your new SQL Server database? Type the name of the new SQL Server database. Access revises the name if it conflicts with an existing database name and adds a numbered suffix (mydatabase 1, for example).

[About ODBC data sources](#)

An [ODBC data source](#)— for example, a database and the server on which it resides— is accessed through an [Open Database Connectivity](#) (ODBC) driver.

[What is a data source?](#)

A data source is a source of data and the connection information needed to access that data. Examples of data sources are Microsoft Access, Microsoft SQL Server, Oracle RDBMS, a spreadsheet, and a text file. Examples of connection information include server location, database name, logon ID, password, and various ODBC driver options that describe how to connect to the data source.

In the ODBC architecture, an application (such as Access or a Microsoft Visual Basic program) connects to the ODBC Driver Manager, which in turn uses a specific ODBC driver (for example, Microsoft SQL ODBC driver) to connect to a data source (in this case, a [Microsoft SQL Server database](#)). In Access, you use ODBC data sources to connect to data sources external to Access that do not

have built-in drivers.

To connect to these data sources, you must do the following:

- Install the appropriate ODBC driver on the computer that contains the data source.
- Define a data source name (DSN) by using either the **ODBC Data Source Administrator** to store the connection information in the Microsoft Windows registry or a DSN file, or a connect string in Visual Basic code to pass the connection information directly to the ODBC Driver Manager.

[Machine data sources](#)

Machine data sources store connection information in the Windows Registry on a specific computer with a user-defined name. You can use machine data sources only on the computer they are defined on. There are two types of machine data sources— user and system. User data sources can be used only by the current user and are visible only to that user. System data sources can be used by all users on a computer and are visible to all users on the computer and system-wide services. A machine data source is especially useful when you want to provide added security, because only users who are logged on can view a machine data source and it cannot be copied by a remote user to another computer.

[File data sources](#)

File data sources (also called DSN files) store connection information in a text file, not the Windows registry, and are generally more flexible to use than machine data sources. For example you can copy a file data source to any computer with the correct ODBC driver so that your application can rely on consistent and accurate connection information to all the computers it uses. Or you can place the file data source on a single server, share it between many computers on the network, and easily maintain the connection information in one location.

A file data source can also be unshareable. An unshareable file data source resides on a single computer and points to a machine data source. You can use unshareable file data sources to access existing machine data sources from file data sources.

[Connect strings](#)

If you are a programmer, you can define a formatted connect string in your Microsoft Visual Basic code that specifies the connection information. Using a connect string avoids defining a machine or file DSN and passes the connection information directly to the ODBC Driver Manager. This is useful, for example, when you want to avoid requiring system administrators or users to first create a DSN, or to simplify the installation of your application. To help protect the connection string information in your code, employ security methods for your code by using a MDE file or a password.

[Enter database and transaction log devices and sizes for a Microsoft SQL Server version 6.5 database](#)

If you choose to create a new Microsoft SQL Server 6.5 database, the Upsizing Wizard asks you to enter devices and sizes for your database and transaction log. It also asks you to enter a pre-allocated size for the database and transactions.

[Enter a database device and size](#)

All Microsoft SQL Server databases and transaction logs are placed on devices. A device is both a physical file and a logical location for databases and transaction logs. When you create a device, SQL Server creates a file and reserves a set amount of disk space for its own use.

If you are a system administrator, the Upsizing Wizard displays the amount of free space available on the existing devices. Select a device that has at least enough free space for your estimated database size.

If no existing devices have enough free space, you can create a new device using the Upsizing Wizard. New devices should be at least as big as your estimated database size. If possible, make the device larger than your database size, so you can expand your database later, or place other databases or transaction logs on the same device.

[Enter a transaction log device and size](#)

The transaction log is a chronological record of all activity that occurs on the

server, SQL Server uses this log for database backup and recovery.

The transaction log device you select must have at least one megabyte of space available. Ideally, a database and its corresponding transaction log should be placed on devices that are on separate physical disks. These devices must be created before starting the Upsizing Wizard, because the Upsizing Wizard creates all new devices on the same physical disk where the Master database device resides.

If you have only one physical disk, you should place the database and its transaction log on separate devices, so that the log will be available even if the database encounters problems.

The Upsizing Wizard creates the transaction log using the size you specify. The size of your transaction log should be at least as large as the size of all of the upsized tables in your database. The size you specify can't exceed the space available on the transaction log device you selected.

Note For new databases, the Upsizing Wizard truncates the transaction log file on checkpoint if it fills up before the upsizing is complete. However, it's possible for the transaction log to overflow on a single table. If this happens, Microsoft Access doesn't upsize the table and writes the error to the Upsizing Wizard report.

[Enter a database and transaction log size](#)

When SQL Server 6.5 creates a database, it sets aside a fixed amount of space for that database on one or more devices. Not all of this space is necessarily used by the database right away. The database size is the maximum size a database can become before it runs out of space.

To estimate the space needed for your database, look at the size of your Access database and estimate the rate at which your new SQL Server database will grow. In general, every megabyte of Access data requires at least 1.3 to 1.5 megabytes on SQL Server. Note that the size of a Access database (.mdb) file includes all Access database objects, not just data.

If you have ample disk space on your server, multiply the size of your Access database by two. This ensures that the Upsizing Wizard has enough space to

upsized your database and also leave it some room to grow. If you expect to add a lot of data to the database, make the multiple larger.

The transaction log size should be about 25% of the database size.

[Choose which tables to upsize](#)

You can select the [Microsoft Access database](#) tables you want to upsize to a Microsoft SQL Server database.

[Available Tables / Export to SQL Server](#)

The **Available Tables** list box includes all linked tables except for SQL Server tables already in an SQL Server database. Linked tables that point to an SQL Server database that has been selected for upsizing automatically appear in the **Export to SQL Server** list box and can't be removed. Any table that has a name ending in "_local" is excluded from the list of available tables to prevent upsizing tables that have already been upsized. If you do want to upsize these tables again, rename them before you run the Upsizing Wizard by removing the suffix "_local". Tables that are not currently visible in the [Database window](#) are also excluded, including hidden tables and system tables.

[Why aliasing queries may be created in your Access database](#)

If you choose to link SQL Server tables to your Access database, the Upsizing Wizard may create an aliasing query to handle name differences between Access and SQL Server version 6.5, which doesn't allow spaces or special characters in field names other than pound sign (#), dollar sign (\$), and underscore (_) characters. The Upsizing Wizard automatically replaces spaces and illegal characters with the "_" character. If field names are changed when a table is upsized, the Upsizing Wizard names the linked table with the suffix "_remote" (for example, Employees_remote) and renames the original table by adding the suffix "_local" (for example, Employees_local). The Upsizing Wizard then creates an aliasing query called Employees so that forms, reports, and queries continue to work properly on the new linked SQL Server table.

[The effect of Declarative Referential Integrity \(DRI\) on table upsizing](#)

If you have previously selected Declarative Referential Integrity (DRI) to upsize tables, DRI prevents you from overwriting related tables that were previously upsized. You can drop an upsized SQL Server database by using the **Drop SQL Database** command (**Tools** menu, **Database Utilities** submenu) in SQL Server 2000 Desktop Engine, or by using SQL Server's Enterprise Manager, and then upsizing the Access database again. If you want to re-upsize only some tables, you must delete those tables and any other tables they are related to starting with the "many" table in a [one-to-many relationship](#) (a table that has no references to its primary key from another table), and then upsize the entire set of tables again. The Upsizing Wizard can only establish DRI relationships between related tables that are upsized at the same time, but it can't establish relationships between those tables and pre-existing tables on the server.

[Choose which table attributes to upsize](#)

You can select which table attributes to upsize to your Microsoft SQL Server database. At a minimum, the Upsizing Wizard converts Microsoft Access field names to legal SQL Server field names, and converts Access data types to the equivalent SQL Server data types. By default, all attributes are selected for upsizing.

[What table attributes do you want to upsize?](#)

[Indexes](#)

If you select the **Indexes** check box, the Upsizing Wizard upsizes all indexes.

The Upsizing Wizard converts Microsoft Access primary keys to Microsoft SQL Server non-clustered, unique indexes and marks them as SQL Server primary keys. If you choose to link the upsized SQL Server table to your Access database, the Upsizing Wizard also adds the prefix "aaaaa." Microsoft Access chooses the index that is first alphabetically in the list of available indexes as the primary key and the "aaaaa" prefix ensures that the right index is chosen.

All other indexes retain their names, except where illegal characters are replaced with the "_" character. Unique and non-unique Microsoft Access indexes become unique and non-unique SQL Server indexes. Note that SQL Server doesn't support ascending or descending indexes.

A linked table must have a unique index to be updateable in Microsoft Access. The Upsizing Wizard can upsize an existing unique index, but can't create one where none exists. If you want to be able to update your tables, make sure you add a unique index to each Access table before upsizing.

[Validation rules](#)

Validation rules

If you select the **Validation rules** check box, the Upsizing Wizard upsizes all field **Required** properties and table, record, and field validation rules as update and insert [triggers](#).

A trigger is a series of Transact-SQL statements associated with an SQL Server table. A table can have three triggers, one for each of the commands that can modify data in a table: the UPDATE, INSERT, and DELETE commands. The trigger is automatically executed when the command is carried out. The Upsizing Wizard uses triggers rather than SQL Server rules to enforce field level validation because SQL Server rules do not allow you to display custom error messages.

Each validation rule doesn't necessarily have a one-to-one correspondence with a trigger. Each validation rule may become part of several triggers, and each trigger may contain code to emulate the functionality of several validation rules.

When you set the **Required** property of a Microsoft Access field to true, a user cannot insert a record and leave the required field null (if there is no default bound to the field) or make the field null when updating a record. Required fields are upsized to fields that don't allow [Null](#) values on SQL Server.

Validation text

For SQL Server 2000 databases, the Access database **Validation Text** property is converted to the Access project **Validation Text** property. This enables the Access friendly error messages to be displayed in the event of a constraint violation at run time.

For SQL Server 7.0 and 6.5 databases, the Upsizing Wizard creates a fragment of Transact-SQL code that presents a message similar to the one Microsoft

Access displays if the field is null when a record is added or updated.

[Defaults](#)

If you select the **Defaults** check box, the Upsizing Wizard will upsize all **Default Value** properties to [American National Standards Institute \(ANSI\)](#) default objects.

[Table relationships](#)

You can decide how to upsize table relationships and referential integrity by using update, insert, or delete triggers, or by using Declared Referential Integrity (DRI). DRI works the same way as Microsoft Access referential integrity by defining primary key constraints for base tables (the "one" side of a one-to-many relationship) and foreign key constraints for foreign tables (typically the "many" side of a one-to-many relationship).

- **Use triggers** If your Microsoft Access table relationships have cascading updates or deletes defined, you may want to preserve this behavior in upsized tables. To do this, select the **Table relationships** check box and **Use triggers** radio button to upsize any cascading updates or deletes as triggers to enforce referential integrity.

A table relation doesn't necessarily have a one-to-one correspondence with a trigger. Each relation may become part of several triggers or each trigger may contain code to emulate the functionality of several referential integrity rules. Insert triggers are used on child tables and delete triggers on parent tables.

Note Access requires DRI to read the [database diagram](#) of an SQL Server database. To allow Access to read the database diagram, while enforcing referential integrity through triggers, the Upsizing Wizard places DRI on the relationships but turns off checking of foreign key constraints.

- **Use DRI** Select the **Table relationships** check box and **Use DRI** radio button, to use DRI to enforce referential integrity. The Upsizing Wizard performs differently depending on the version of SQL Server you are upsizing to.

SQL Server 2000 database If you select **Use DRI**, then Access database column validation is converted to a SQL Server DRI check constraint with a validation message as the following table shows.

Access database relationship	SQL Server Foreign Key	Validation Text
Cascade Update	ON DELETE NO ACTION ON UPDATE CASCADE	"The record cannot be deleted because the table <foreigntable> includes related records."
Cascade Delete	ON DELETE CASCADE ON UPDATE NO ACTION	"You cannot add or change a record because a related record is required in table <primarytable>."
Cascade Update and Cascade Delete	ON DELETE CASCADE ON UPDATE CASCADE	Value is not set.
No DRI	ON DELETE NO ACTION ON UPDATE NO ACTION	You cannot add, change, or delete this record because of the relationship restriction between the tables <primarytable> and <foreigntable>.

SQL Server version 7.0 and 6.5 databases These databases do not support cascading update or deletes. Your Access database may not have cascading updates or deletes defined, or you don't care if you lose these features in the upsized database. The Upsizing Wizard warns you that you will lose the cascading update and delete functionality.

[What data options do you want to include?](#)

[Add timestamp fields to tables](#)

Microsoft SQL Server uses a timestamp field to indicate that a record was changed (but not when it was changed) by creating a unique value field and then updating this field whenever a record is updated. For a linked table, Access uses the value in timestamp fields to determine whether a record has been changed

before updating it. In general, a timestamp field provides the best performance and reliability. Without a timestamp field, SQL Server must check all the fields in the record to determine if the record has changed, which slows performance.

- If you select the default value, **Yes, let wizard decide**, the Upsizing Wizard creates new fields with the timestamp data type in SQL Server tables that have been upsized from Microsoft Access tables that contain floating-point (Single or Double), [Memo](#), or [OLE object](#) fields.
- You can also choose to have the Upsizing Wizard create a timestamp field for all upsized tables regardless of what field types they contain by selecting **Yes, always**. This improves the performance of upsized Access tables that may not contain Memo, OLE Object, or floating-point fields, but have fields of other types.
- If you select the **No, Never** option, the Upsizing Wizard does not add timestamp fields to tables.

Note In linked SQL Server tables, Microsoft Access doesn't check to determine if Memo or OLE object fields have changed because these fields could be many megabytes in size and the comparison could be too network-intensive and time-consuming. Therefore, if only a text or image field has changed and there is no timestamp field, Microsoft Access overwrites the change. Also, the value of a floating-point field may appear to have changed when it hasn't, so in the absence of a timestamp field, Microsoft Access may determine that the record has been changed when it has not.

[Only create the table structure, don't upsize any data](#)

The Upsizing Wizard upsizes all data to SQL Server by default. If you select the **Only create table structure, don't upsize any data** check box, only the data structure is upsized.

[Choose how to upsize your application](#)

You can select different ways to upsize your [Microsoft Access database](#) application.

[What application changes do you want to make?](#)

[Create a new Access client/server application](#)

If you select **Create new Access client/server application**, the Upsizing Wizard creates a new [Microsoft Access project](#). The Upsizing Wizard prompts you for a name, which defaults to the current Access database name, adds a "CS" suffix, and then stores the project in the same location as the existing Access database.

The Upsizing Wizard creates the Access project file and then upsizes all the database objects from the Access database to the Access project. If you don't save the password and user ID, then the first time you open the Access project, Access displays the **Data Link Properties** dialog box so that you can connect to an SQL Server database.

[How database objects are upsized](#)

The following data and database objects get upsized:

[Data and data types](#)

For SQL Server version 7.0 or later databases, the Upsizing Wizard converts Access database text to Unicode by adding the Unicode string identifier to all string values and by adding the Unicode n prefix to all data types.

All Access database data types are converted to their equivalent in SQL Server.

[Queries](#)

For SQL Server 2000 databases:

- Select queries that don't have an ORDER BY clause or parameters are converted to views.
- Action queries are converted to stored procedure action queries. Access adds SET NOCOUNT ON after the parameter declaration code to make sure the stored procedure runs.
- Select queries that only reference tables (also called base queries) that use either parameters or an ORDER BY clause are converted to user-defined functions. If necessary, the TOP 100 PERCENT clause is added to a query that contains an ORDER BY clause.

- Parameter queries that use named parameters maintain the original text name used in the Access database and are converted either to stored procedures or inline user-defined functions.

For SQL server 7.0 and 6.5 databases:

- Select queries are converted to views.
- Sorted queries are converted to a combination of views and stored procedures to allow for nesting and sorting (Views can be nested but can't contain ORDER BY clauses; stored procedures can contain ORDER BY clauses but can't be nested).
- Parameter queries, queries that depend on a parameter query, and action queries are converted to stored procedures.

Note For all SQL Server databases, you may need to manually convert queries that did not upsize, such as SQL Passthrough queries, data definition queries, and crosstab queries. You may also have to manually upsize queries that were nested too deeply.

[Forms, reports, and controls](#)

For all SQL Server databases, SQL statements in **RecordSource**, **ControlsSource** and **RowSource** properties for forms, reports, or controls are kept in place and not converted to a stored procedure or user-defined function.

[Data access pages](#)

The Upsizing Wizard changes the OLE DB connection and the data binding information in the Microsoft Office data source control to work with the new SQL Server database, and copies the page's corresponding HTML file to the same location as the Access project, renaming the HTML file with a "_CS" suffix. The new pages in the Access project retain the original name, so that hyperlinks between the Access project Pages continue to work.

[Customized command bars and startup properties](#)

For all versions of SQL Server, the Upsizing Wizard upsizes any custom command bars as well as the following startup properties:

StartUpShowDBWindow
StartUpShowStatusBar
AllowShortcutMenus
AllowFullMenus
AllowBuiltInToolbars
AllowToolbarChanges
AllowSpecialKeys
UseAppIconForFrmRpt
AppIcon
AppTitle
StartUpForm
StartUpMenuBar
StartupShortcutMenuBar

[Modules and macros](#)

The Upsizing Wizard doesn't make any changes to modules and macro. Designing and building an optimized client/server application is different from designing and building an optimized file server application. Therefore, you will probably need to retrofit your application to take full advantage of SQL Server and an Access project. After you upsize, you need to manually convert code that uses recordsets from [Data Access Objects \(DAO\)](#) to [ActiveX Data Objects \(ADO\)](#) in your modules. You should also revise any table and query design code (The Upsizing Wizard does not upsize an SQL Data Definition Language). For more information on developing optimized client/server applications, see the *Microsoft Office 2000/Visual Basic Programmer's Guide*.

[Link SQL Server tables to existing application](#)

If you select **Link SQL Server tables to existing application**, the Upsizing Wizard can modify your Access database so that your queries, forms, reports, and data access pages use the data in the new Microsoft SQL Server database rather than the data in your Microsoft Access database.

The Upsizing Wizard renames the Microsoft Access tables you upsize with the suffix "_local." For example, if you upsize a table named Employees, the table is renamed Employees_local in your Access database. Then, the Upsizing wizard creates a linked SQL Server table named Employees.

Queries, forms, reports, and data access pages based on the original Employees tables will now use the linked SQL Server Employees table. Many of the properties of the fields in the original local table are inherited by the new local table including, **Description**, **Caption**, **Format**, **InputMask**, and **DecimalPlaces**.

[No application changes](#)

Select **No application changes** if you only want to upsize your data and not make any other changes to your existing Access database application.

[Save password and user ID](#)

By default, the Upsizing Wizard creates linked tables in the existing application or creates an Access project without saving the username and password. This means that users are prompted for a username and password each time they log on to an SQL Server database.

If you select **Save password and user ID**, users can access an SQL Server database without logging in. If you select **Create new Access client/server application**, the Access project stores the username password in the OLE DB connection string.

Note This option is disabled for the **No application changes** option if a linked SQL Server table is configured with an **MSysConf** table to deny saving passwords.

[Troubleshoot the Upsizing Wizard](#)

Note The Upsizing Wizard creates a report that provides a detailed description of all objects created, and reports any errors encountered during the process. The Upsizing Wizard automatically creates this report as a [report snapshot](#) with the same name as the Access database (for example, Northwind.snp), and stores it in the default database folder.

[I can't find my switchboard manager form.](#)

The Upsizing Wizard doesn't upsize a switchboard manager form. The Switchboard Manager uses a table to store information about the form and is not

designed for a client-server environment.

[I can't upsize my MDE file.](#)

When you save a Microsoft Access database as an MDE file, all modules and source code within the database are compiled and removed. Your Visual Basic for Applications code will continue to run, but you can't view it or edit it. If you run upsizing tools with the **Create a new Access client/server application** option, the Upsizing Wizard must have access to objects that were removed when you saved the database as an MDE file.

You may select either the **No application changes** or **Link SQL Server tables to an existing application** options when upsizing an MDE, but if you want to create a client-server application, you must use the original Access database (.mdb) that the MDE file was created from.

[My DDE code doesn't work in my Access project](#)

[DDE](#) code is not supported in an Access project. If you upsize an Access database to an Access project, you will need to convert the DDE code to ADO code.

[My upsized table contains no records.](#)

A field in the table in the Access database may contain a null value for more than one record, but the **Indexed** property is set to **Yes (No Duplicates)** and the **Required** property is set to **No**. If a field in your table is indexed using these property settings, do not add more than one record that contains a null value to the column.

Remove the records with the null values and then upsize the table again.

[I ran out of disk space and my SQL Server database was not completely upsized.](#)

If the Upsizing Wizard runs out of disk space on the server, the wizard halts, leaving a partial database and any devices it created on the server. If this happens, you can drop databases by using the **Drop Database** command (**Tools** menu, **Database Utilities** submenu) in SQL Server 2000 Desktop Engine, or by

using SQL Server Enterprise Manager. For more information on SQL Server Enterprise Manager, see the SQL Server documentation.

[I'm having problems accessing or creating database objects](#)

You may not have the correct permissions. You need read/design permission on all Access database objects to upsize them. You need the following permissions in a SQL Server database:

- To upsize to an existing database, you need CREATE TABLE and CREATE DEFAULT permissions.
- To build a new database, you need CREATE DATABASE permission, and SELECT permissions on the system tables in the Master database.
- To create new devices, you must be a system administrator.

[I can't upsize my data access pages](#)

If the pages in your Access database are located on a web site, then the Upsizing Wizard can't copy them. To upsize these pages:

1. Copy all page files and related HTML files to your local computer.
2. Open each page in the database window and change its file location.
3. Run the Upsizing Wizard.
4. Test your pages in the new Access project.
5. Move the pages and all related HTML files back to the original Web site.

[Show All](#)

Troubleshoot the Upsizing Wizard

Note The Upsizing Wizard creates a report that provides a detailed description of all objects created, and reports any errors encountered during the process. The Upsizing Wizard automatically creates this report as a [report snapshot](#) with the same name as the Access database (for example, Northwind.snp), and stores it in the default database folder.

[I can't find my switchboard manager form.](#)

The Upsizing Wizard doesn't upsize a switchboard manager form. The Switchboard Manager uses a table to store information about the form and is not designed for a client-server environment.

[I can't upsize my MDE file.](#)

When you save a Microsoft Access database as an MDE file, all modules and source code within the database are compiled and removed. Your Visual Basic for Applications code will continue to run, but you can't view it or edit it. If you run upsizing tools with the **Create a new Access client/server application** option, the Upsizing Wizard must have access to objects that were removed when you saved the database as an MDE file.

You may select either the **No application changes** or **Link SQL Server tables to an existing application** options when upsizing an MDE, but if you want to create a client-server application, you must use the original Access database (.mdb) that the MDE file was created from.

[My DDE code doesn't work in my Access project](#)

[DDE](#) code is not supported in an Access project. If you upsize an Access database to an Access project, you will need to convert the DDE code to ADO code.

[My upsized table contains no records.](#)

A field in the table in the Access database may contain a null value for more than one record, but the **Indexed** property is set to **Yes (No Duplicates)** and the **Required** property is set to **No**. If a field in your table is indexed using these property settings, do not add more than one record that contains a null value to the column.

Remove the records with the null values and then upsize the table again.

[I ran out of disk space and my SQL Server database was not completely upsized.](#)

If the Upsizing Wizard runs out of disk space on the server, the wizard halts, leaving a partial database and any devices it created on the server. If this happens, you can drop databases by using the **Drop Database** command (**Tools** menu, **Database Utilities** submenu) in SQL Server 2000 Desktop Engine, or by using SQL Server Enterprise Manager. For more information on SQL Server Enterprise Manager, see the SQL Server documentation.

[I'm having problems accessing or creating database objects](#)

You may not have the correct permissions. You need read/design permission on all Access database objects to upsize them. You need the following permissions in a SQL Server database:

- To upsize to an existing database, you need CREATE TABLE and CREATE DEFAULT permissions.
- To build a new database, you need CREATE DATABASE permission, and SELECT permissions on the system tables in the Master database.
- To create new devices, you must be a system administrator.

[I can't upsize my data access pages](#)

If the pages in your Access database are located on a web site, then the Upsizing Wizard can't copy them. To upsize these pages:

1. Copy all page files and related HTML files to your local computer.
2. Open each page in the database window and change its file location.
3. Run the Upsizing Wizard.
4. Test your pages in the new Access project.

5. Move the pages and all related HTML files back to the original Web site.

[Show All](#)

About columns and column properties (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Columns](#)

In a table, data is arranged into columns. Each column stores one data element, such as a first name, one line of an address, a price, or any similar discrete unit of information.

When columns are created in a table, they are given a name that identifies their purpose, such as `FirstName` or `Address1`. In most databases, you must also specify additional properties, such as how long the longest entry in the column will be, and what type of data the column will contain - characters, integers, floating-point numbers, dates or times, and so on. Other column properties can include whether the column is the table's primary key, whether users must enter a value into it, and what its default value is.

[Column properties](#)

Each column in a table has a set of properties. Each property defines one characteristic, such as the name, data type or length, of a column. The entire set of properties for a column makes up that column's definition in your database table.

You can set column properties directly in your database table in a database diagram. Three column properties are required— the column name, data type, and length— before you can save a table in your database. You can redefine a column by editing any of its properties. For example, you can rename a column, alter its length, specify a default value, and so on.

When you view your table in column properties view, the column properties are laid out in a grid. In this grid, you can choose which properties you want to see.

You can hide the properties that you rarely define and show the properties that you define more frequently. You can also resize the columns in the grid so that the properties are easy to read.

publishers							
	Column Name	Datatype	Length	Precision	Scale	Allow Nulls	▲
🔍▶	pub_id	char	4	0	0	<input type="checkbox"/>	
	pub_name	varchar	40	0	0	<input checked="" type="checkbox"/>	
	city	varchar	20	0	0	<input checked="" type="checkbox"/>	
	state	char	2	0	0	<input checked="" type="checkbox"/>	
	country	varchar	30	0	0	<input checked="" type="checkbox"/>	
						<input type="checkbox"/>	▼

[Show All](#)

Set column properties (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can set or change the properties of a column.

[Set column properties in a database diagram](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose column properties you want to define.
3. Right-click the table and choose **Properties** from the shortcut menu.
4. Select the **Columns** tab.
5. For each property you want to change, modify the value or setting.
6. Close the **Properties** dialog box.

The new setting takes effect in the database diagram as soon as you press the TAB key or click outside the cell in the **Properties** dialog box. The new setting is saved in the database when you save the table or your database diagram.

Note If a column property is visible on the database diagram, you can edit that property value directly— without opening the property page.

[Set column properties in Table Designer](#)

1. Click **Tables**  under **Objects**, click the table you want to open, and then click **Design** on the Database window toolbar.
2. To set a property, click in the grid cell for that property of the database column you want to modify, then enter the value.
-or-
Select the grid row describing the database column you want to modify, then modify the property in the tab appearing below the grid.

The new setting takes effect in the database diagram as soon as you press the TAB key or click outside the edited grid cell or control. The new setting is saved in the database when you save the table or your database diagram.

[Show All](#)

Troubleshoot tables and columns (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[I get the message "Error validating the formula for column."](#)

The formula in the calculated column contains an error. Common formula errors include:

- Mismatched parentheses.
- Misspelled column names.
- Using an operator on a column whose data type does not support that operator.

[I can't create or modify an identity column.](#)

An identity column contains system-generated, sequential values that uniquely identify each row within a table. Keep in mind the following guidelines when creating an identity column:

- An identity column must have a datatype of bigint, int, smallint, tinyint, decimal, or numeric with a scale of 0.
- An identity column cannot have [Null](#) values. All identity columns must contain unique values other than Null. To make the column an identity column, first clear the **Allow Nulls** check box on the **Column Property** dialog box.
- An identity column cannot have default values.
- The identity increment or seed must be a number that can be stored in the data type size you have chosen. If the data type of the column is **tinyint**, you must enter a positive number. For any other data type, you can enter a positive or negative number.

[I'm having trouble creating and modifying tables.](#)

- A table must have at least one column that is not a computed column. You can, however, create a view containing all computed columns.
- You can only have one copy of any table on a diagram. If you want to create a new table that has some of the same columns as an existing table, you can duplicate an existing table as the first step in creating a new table.
- The table being loaded into memory has a user-defined data type that is not recognized. The table you are loading into memory refers to a recently created user-defined data type, but your in-memory list of user-defined data types does not include it. Because the data type was created recently, it is not present in local memory. All user-defined data types are loaded from the database into local memory when you open the first Table Designer window or database diagram within your connection to that database. To refresh your in-memory copy of the database's user-defined data types, close all open database diagrams and Table Designer windows within that database connection. After you close them all, reopen them in turn. When you reopen the first one, the list of user-defined data types will be refreshed.
- The contents of the database and the in-memory contents of the Table Designer may be in conflict. This conflict may occur because another user has recently modified the table you are working on, but the Table Designer retains the older version of the table in your local memory. The Table Designer needs to load information about the other table to which you are trying to create a relationship. For example, it needs to load the details about the other table's columns, so you can choose which columns from the table you are designing correspond to columns in the other table. Before proceeding to load the other table into your local memory, the Table Designer will reconcile these conflicts accordingly. The Table Designer does not reconcile all differences between your local memory and the database. It reconciles only those differences that would otherwise prevent you from creating the relationship you are trying to create.

[I'm having problems using the bit data type, it doesn't work the same as the Yes/No data type.](#)

Microsoft Access stores either a zero (0) or a minus one (-1) in the Access database [Yes/No data type](#). An Access database interprets 0 as a 0, No, or False value, and interprets anything else as a -1, Yes, or True value.

By contrast, SQL Server stores either a zero (0) or a plus one (+1) in the [bit data type](#). An Access project interprets 0 as a False value, and interprets plus one (+1)

as a True value.

[Show All](#)

Change a column's identity properties (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table in which you want to change the identity properties of a column.
3. If necessary, right-click the table and select **Custom View** from the shortcut menu.
4. Make sure the **Identity**, **Identity Seed**, and **Identity Increment** properties are visible.

[How?](#)

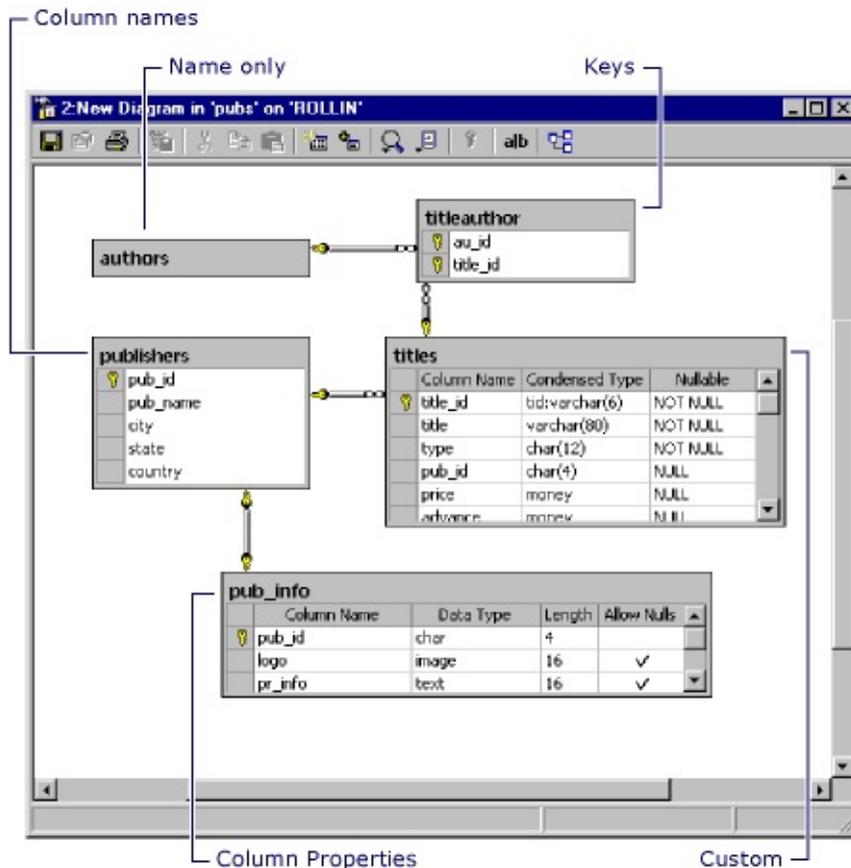
[Background information](#)

When you are working with only a few tables in a database diagram, it is usually helpful to view as much column information as possible. Such a view not only gives you more information, but it also enables you to edit the column definitions.

When you are working with a large number of tables, however, it is usually preferable to view only the column names, or only the table name, so that you can see more of the tables in your diagram at once.

In a database diagram, you can expand the tables you want to edit as needed and then collapse them again to save space on the diagram.

Each table in a database diagram can be displayed in one of five views.



- **Column Properties**  Shows every database column of the table, but shows only the column name and other fundamental properties of each database column. You can edit the column properties directly by typing in a cell or using the **Cut**, **Copy**, and **Paste** commands. You can also move and delete entire database column definitions.

When you insert a new table into a database diagram, it is shown in this view so that you can begin defining each column.

- **Column Names**  Shows only the name of each database column. You can rename the database columns in this view. When you add an existing table to a diagram, it is shown in this view.
- **Keys**  Shows only the names of only some of the table's columns. The columns whose names appear are those participating in a primary key, a foreign key, or a unique constraint. You can rename some of the database columns in this view.
- **Name Only**  Shows only the table name.
- **Custom**  Shows all of the table's database columns, but shows only

the properties you choose. This gives you the flexibility to display only the information you want. This view is also useful for printing the database diagrams, because you can display more tables on one page, and will not normally see horizontal scroll bars on the tables.

The default set of properties for the Custom view includes Column Name, Condensed Datatype, and Nullable.

Property	Setting
Column Name	The name of the field in the table.
Condensed Type	Information about the field's data type, in the same format as the SQL Create Table statement. For example, a field containing a variable-length string with a maximum length of 20 characters would be represented as <code>varchar(20)</code> .
Nullable	Null (the field can contain a Null value) or Not Null (the field must contain a non-null value).

[Change a table view](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the tables you want to see in another view. To change the view of all the tables in your diagram, choose **Select All** from the **Edit** menu.
3. Right-click the table and select the desired view from the shortcut menu.

When you finish editing a table, you can change it to a different view or resize it to create more space on the database diagram.

[Modify the Custom view](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table you want to see in Custom

view.

3. Click the **Diagram** menu, and then click **Modify Custom View**. The **Column Selection** dialog box appears.
4. Use the arrows to move the desired columns from the **Available columns** box to the **Selected columns** box.

Note To add a column, highlight the column in the **Available columns** box and click the > button. The column will then be listed in the **Selected columns** box. To display all columns, click the >> button. To remove a column from the **Selected columns** box, highlight the column and click the < button.

5. The columns will appear in the table in the order they are displayed in the **Selected columns** box. Use the Sort arrows to rearrange the columns in the **Selected columns** box to the desired order.
6. To make this setting the default Custom view, select the **Save as default** box.

Note If the **Save as default** box is not selected, the columns listed in the **Selected columns** box will be displayed for all the database tables displayed in Custom view on the current diagram. However, this setting will not be saved and is not applied to other diagrams when you choose the **Custom view** command.

The selected table is redrawn to display the columns chosen in the dialog box.

5. If the **Allow Nulls** property is selected, clear the check box.
6. Select the **Identity** cell for the column whose values you want to automatically increment.

Note Only one column per table can be defined as an identity column.

7. Type a value in the **Identity Seed** cell. This value will be assigned to the first row in the table. If you leave this cell blank, the value 1 will be assigned by default.
8. Type a value in the **Identity Increment** cell. This value is the increment that will be added to the **Identity Seed** for each subsequent row. If you leave this cell blank, the value 1 will be assigned by default.

About Table Designer (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The Table Designer is a visual tool allowing you to design and visualize a single table in a database to which you are connected.

The Table Designer has two parts. The upper part shows a grid; each row of the grid describes one database column. For each database column, the grid displays its fundamental characteristics: column name, data type, length, and nulls-allowed setting.

The lower portion of the Table Designer shows additional characteristics for whichever data column is highlighted in the upper portion.

From the Table Designer, you can also access property pages through which you can create and modify relationships, constraints, indexes, and keys for the table.

[Show All](#)

Add tables to a database diagram or database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can add a table to your database diagram to edit its structure or relate it to other tables in your diagram. You can either add existing database tables to a database diagram or insert a new table that has not yet been defined in the database. Alternatively, you can create a table or modify an existing table with the Table Designer.



[Insert a new table in a database diagram](#)

Adding a new table to the database diagram means that you are defining a new table that does not already exist in the database. To create a new table, you must define the individual columns that make up the table. The table is created in the database when you save the table or the database diagram.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.

To create a table in your current diagram, click **New Table** on the **Insert** menu.

–or–

Right-click in the diagram and select **New Table**.

2. To create a table in your current diagram, click **New Table** on the **Insert** menu.

–or–

Right-click in the diagram and select **New Table**.

3. Modify or accept the system-assigned table name, in the **Choose Name** dialog box, and then choose **OK**.
A new table appears in the diagram in column properties view.
4. In the first cell of the new table, type a column name. Then press the TAB key to move to the next cell.
5. Under **Data Type**, select a data type for the column. Each column must have a name and data type.
You can set the column's other properties, such as Scale, by using the **Columns** property page.
6. Repeat steps 3 and 4 for each column you want to add to the table.

When you save your database diagram, the new table will be added to your database.

Note If you create a new table, then remove it from the database diagram before saving it to the database, the table name remains in memory until you close the database diagram. Also, if you delete an existing table from the database, the table name remains in memory. To use the table name again, close and restart the Database Designer.

[Add existing tables to a database diagram](#)

Adding an existing table means that the table you want to appear in your diagram already exists in your database. If you add a group of tables to a diagram, any relationships that exist between the tables are also added to the diagram.

1. Right click on the database diagram, and then click **Add Table**
.
2. Select the table you want in the **Tables** list, and then click **Add**.
3. Repeat step 2 for each additional table you want to add.
4. Click **Close**.

If relationships exist between the selected table and other tables in your database diagram, relationship lines are automatically drawn.

When you add a table to a database diagram, the definition of the table (not the data that is stored in the table) is loaded from your database into memory. At that point you can edit the table's definition. For example, you can add new columns or modify its indexes, key, relationships, or constraints.

[Add related tables to a database diagram](#)

For tables with existing foreign key constraints, you can easily add the related tables to the database diagram.

1. Select one or more tables with foreign key constraints in the database diagram.
2. Right-click on any of the selected tables and choose **Add Related Tables**.

Both those tables referenced by a foreign key constraint from the selected table(s) and those referencing the selected table(s) with a foreign key constraint are added to the database diagram.

[Show All](#)

Change a table view in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

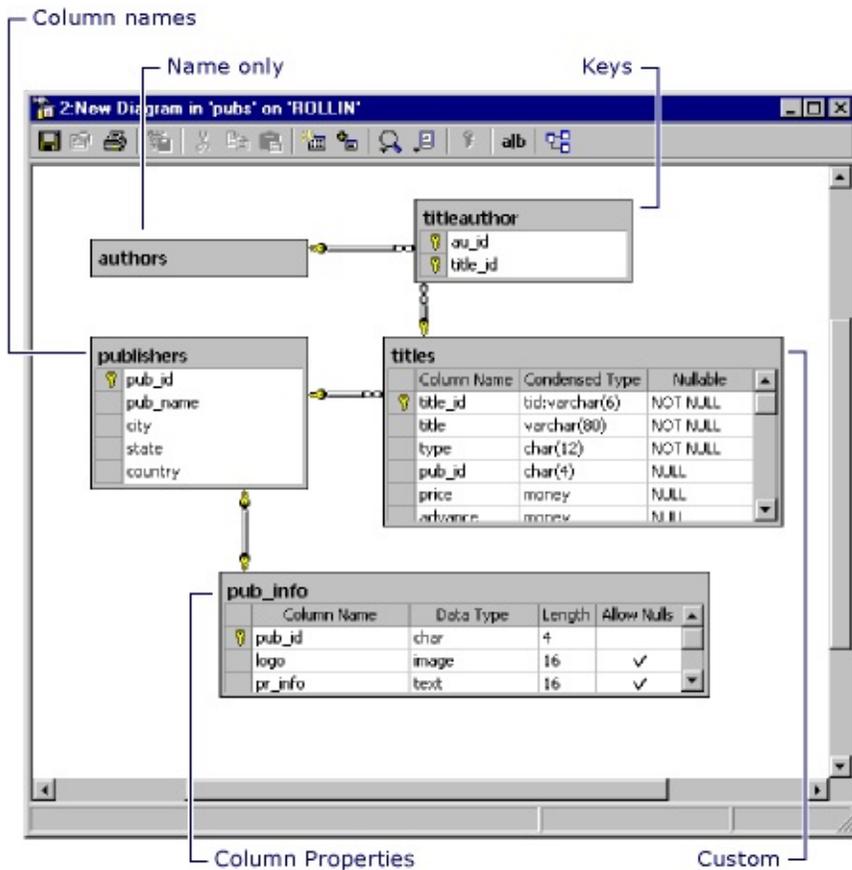
 [Background information](#)

When you are working with only a few tables in a database diagram, it is usually helpful to view as much column information as possible. Such a view not only gives you more information, but it also enables you to edit the column definitions.

When you are working with a large number of tables, however, it is usually preferable to view only the column names, or only the table name, so that you can see more of the tables in your diagram at once.

In a database diagram, you can expand the tables you want to edit as needed and then collapse them again to save space on the diagram.

Each table in a database diagram can be displayed in one of five views.



- **Column Properties**  Shows every database column of the table, but shows only the column name and other fundamental properties of each database column. You can edit the column properties directly by typing in a cell or using the **Cut**, **Copy**, and **Paste** commands. You can also move and delete entire database column definitions.

When you insert a new table into a database diagram, it is shown in this view so that you can begin defining each column.

- **Column Names**  Shows only the name of each database column. You can rename the database columns in this view. When you add an existing table to a diagram, it is shown in this view.
- **Keys**  Shows only the names of only some of the table's columns. The columns whose names appear are those participating in a primary key, a foreign key, or a unique constraint. You can rename some of the database columns in this view.
- **Name Only**  Shows only the table name.
- **Custom**  Shows all of the table's database columns, but shows only the

properties you choose. This gives you the flexibility to display only the information you want. This view is also useful for printing the database diagrams, because you can display more tables on one page, and will not normally see horizontal scroll bars on the tables.

The default set of properties for the Custom view includes Column Name, Condensed Datatype, and Nullable.

Property	Setting
Column Name	The name of the field in the table.
Condensed SQL Type	Information about the field's data type, in the same format as the SQL Create Table statement. For example, a field containing a variable-length string with a maximum length of 20 characters would be represented as <code>varchar(20)</code> .
Nullable	Null (the field can contain a Null value) or Not Null (the field must contain a non-null value).

[Change a table view](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the tables you want to see in another view. To change the view of all the tables in your diagram, choose **Select All** from the **Edit** menu.
3. Right-click the table and select the desired view from the shortcut menu.

When you finish editing a table, you can change it to a different view or resize it to create more space on the database diagram.

[Modify the Custom view](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table you want to see in Custom view.
3. Click the **Diagram** menu, and then click **Modify Custom View**. The

Column Selection dialog box appears.

4. Use the arrows to move the desired columns from the **Available columns** box to the **Selected columns** box.

Note To add a column, highlight the column in the **Available columns** box and click the > button. The column will then be listed in the **Selected columns** box. To display all columns, click the >> button. To remove a column from the **Selected columns** box, highlight the column and click the < button.

5. The columns will appear in the table in the order they are displayed in the **Selected columns** box. Use the Sort arrows to rearrange the columns in the **Selected columns** box to the desired order.
6. To make this setting the default Custom view, select the **Save as default** box.

Note If the **Save as default** box is not selected, the columns listed in the **Selected columns** box will be displayed for all the database tables displayed in Custom view on the current diagram. However, this setting will not be saved and is not applied to other diagrams when you choose the **Custom view** command.

The selected table is redrawn to display the columns chosen in the dialog box.

Hide a table in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can hide a table in your database diagram. Hiding a table does not alter your database. The table and its relationships to other tables continue to exist in the database.

If you hide a table that has been changed either by you or another user, a message prompts you to save the table before hiding it:

- If you save the table before removing it, the database is updated with the changes.
 - If you discard changes to the table before removing it, the table remains in memory in its modified state until you save the table or close the last open database diagram, even though the table no longer appears in your diagram.
1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
 2. In your database diagram, select the table you want to remove.
 3. Right-click the table and choose **Hide Table** from the shortcut menu.

If the table has unsaved changes as a result of edits you made in the database diagram, a message prompts you to save the table before removing it.

The table is hidden from your database diagram but it continues to exist in the database.

[Show All](#)

Change which properties appear in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you show a database table in column properties view, some but not all of that table's property columns are visible. You can customize this view by choosing whether to hide or show each property column in a database table.

Note When you change which properties appear, you change only the displayed properties for that table in that database diagram.

[Hide a property column](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose property column you want to hide. If necessary, right-click the table, and then click **Column Properties** or **Custom View**.
3. Position the pointer along the right border of the column header for the column that you want to hide. The cursor changes from an arrow to a splitter pointer ↔.
4. Drag the column border to the left until it meets the column's left border.
5. Release the mouse button.

[Show a property column that is hidden](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose property column you want to show. If necessary, right-click the table, and then click **Column Properties** or **Custom View**.

3. Position the pointer along the right border of the previous column's header. The cursor changes from an arrow to a splitter pointer ↔.
4. Double-click the splitter pointer. The hidden column resizes to fit the widest entry.

Note You can also set which column properties you want to display for database tables in a database diagram by using the **Custom view** command on the **Table** submenu of the **View** menu.

[Show All](#)

Copy columns from one table to another (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can copy columns from one table to another table in the same database diagram or in different database diagrams. Copying a column involves only the column definition. The data itself is not automatically transferred to the second table as part of this process.



Tip

You can use queries to copy the column's data from the original column to the new column. Use the Query Designer to run an update query to add the data to the table to which you copied the columns.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the columns that you want to copy.
3. Click the **Copy** button on the toolbar. This action copies the selection from the table and places the column and its current set of properties on the Clipboard.
4. Position the cursor in the table at the location where you want to insert the columns.
5. Click the **Paste** button on the toolbar. The column and its properties are inserted at the new location.

When you copy a database column that has a user-defined data type from one database to another, the user-defined data type may not be available in the destination database. In such a case, the column will be assigned the nearest matching base data type available in that database.

Copy a table across database diagrams (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can copy a table from one database diagram to another in the same database.

Copying a table from one database diagram to another database diagram merely adds a reference to the table in the second database diagram. The table is not duplicated in your database. For example, if you copy the authors table from one database diagram to another, each database diagram references the same authors table in the database.

1. For the source and target database diagrams, in the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In the source database diagram, select the table that you want to copy to the target database diagram.
3. Click the **Copy** button on the toolbar. This action places the selected table definition on the Clipboard.
4. Switch to the target database diagram. This database diagram must be in the same database as the source diagram.
5. Click the **Paste** button on the toolbar. The Clipboard contents appear at the new location and remain highlighted until you click elsewhere. If relationships exist between the selected tables and other tables in the target database diagram, relationship lines are automatically drawn.

When you edit the table in either database diagram, your changes are reflected in both database diagrams. Similarly, once you save the table in either database diagram, the table is no longer considered "modified" in either database diagram.

Delete a table from the database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Delete a table from your database when you want to do all of the following:

- Remove the table from your database diagram.
- Remove the table from every other database diagram in which it appears.
- Mark the table for deletion from the database.

A table that is marked for deletion is permanently deleted from the database when you save your database diagram. A reference to that table continues to exist in memory until you save the database diagram. If you close the database diagram without saving it, the table will continue to exist in your database and appear in your database diagram and every other database diagram in which it appeared before you marked it for deletion.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table you want to delete.
3. Right-click the table and choose **Delete Table from Database** from the shortcut menu.
4. A message box prompts you to confirm the deletion. Choose **Yes**.

Note Deleting a table automatically removes any relationships to it.

Delete columns from a table (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Delete columns from a table when they are no longer needed to store data. When a column is deleted from the table in the database diagram, it and all the data it contains are deleted from the database.

Caution This action cannot be undone once the database diagram or table is saved. The only way to restore a deleted column is to close the table or database diagram without saving changes.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the column or columns you want to delete.
3. Right-click the column and choose **Delete Rows** from the shortcut menu.
4. If the column participates in a relationship, a message prompts you to confirm the deletion of the selected columns and their relationships. Choose **OK**.

If the column does not participate in a check constraint, then the column, any constraints attached to it, any relationships it participates in, and any data contained in the column are removed from the database and the database diagram. They are deleted from the database when you save the table or database diagram. If the column does participate in a check constraint, the database server will reject your modification when you save your work. That is, the commit operation will fail. To delete a column that participates in a check constraint, you must first modify or remove the check constraint before deleting the column.

Duplicate a table (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

A table can appear only once in a database diagram. However, if you want to create a new table that contains some or all of the same columns as an existing table in your database diagram, you can duplicate the existing table as the first step in creating your new table. Then in your new table, you can delete unwanted columns, add new columns, and assign a unique table name.

Note This operation duplicates only a table's structure; it does not duplicate any table data.

Use the same technique for duplicating tables across database diagrams.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In the database diagram where you want to duplicate the table, right-click the diagram and choose **New Table** from the shortcut menu.
-or-
Click the **New Table** button  on the Database Diagram toolbar.
3. In the **Choose Name** dialog box, accept or change the system-assigned name of the new table and click **OK**.
A blank table is added to your database diagram.
4. In the table you want to duplicate, select all the columns. This table can be in the same database diagram, a different database diagram in the same database, or a database diagram in a different database.
5. Copy the columns to the blank table.

[Show All](#)

Insert columns into a table (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can add new column definitions to a table to capture additional data that is not already stored in an existing column. The Column Properties view, which shows a table with all of its currently defined columns, provides blank grid rows so that you can easily add new column definitions to your database table.

[Insert columns into a table in a database diagram](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table that you want to add new columns to.
3. If the table is not already in Column Properties or Custom View, right-click the table and choose **Column Properties** or **Custom View** from the shortcut menu.
4. Right-click a row in the table and select **Insert Rows** from the shortcut menu. A blank column row is inserted above the selected row.
-or-
Place the cursor in the first blank cell in the **Column Name** column.
5. In the **Column Name** column, type the column name in the cell. The Column Name is a required value.
6. Press the TAB key to go to the **Data Type** cell. This is a required value.

The default values for your database are added when you create a new column. Define other properties for the column as needed. You can change any of these values and set additional properties at any time according to the rules of your database.

[Insert columns into a table from the Table Designer](#)

1. In the Database window, click **Tables**  under **Objects**, click the table you

- want to open, and then click **Design** on the Database window toolbar.
2. Right-click a row in the table and select **Insert Rows** from the shortcut menu. A blank column row is inserted.

-or-

- Place the cursor in the first blank cell in the **Column Name** column.
3. In the **Column Name** column, type the column name in the cell. The Column Name is a required value.
 4. Press the TAB key to go to the **Data Type** cell. This is a required value.

The default values for your database are added when you create a new column. Define other properties for the column as needed. To define any of the fundamental properties, enter the property value on the grid. To define any of the other properties, highlight the grid row describing that database column, then enter the property value in the appropriate control on the tab appearing under the grid.

[Show All](#)

Move columns from one table to another (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can move columns from one table to another table in the same database diagram or in a different diagram. Moving columns in the database diagram moves the column definition only. The data itself is not automatically transferred to the second table as part of this process.

[Move columns without data from one table to another](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the columns that you want to move.
3. Click the **Cut** button on the toolbar. This action deletes the selection from the table and places the column and its current set of properties on the Clipboard.
4. Position the cursor in the new table at the location where you want to insert the columns.
5. Click the **Paste** button on the toolbar. The columns are inserted at the new location and remain highlighted until you click elsewhere.

[Move columns with data from one table to another](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the columns that you want to move.
3. Click the **Copy** button on the toolbar. This action places the selection with its current set of properties on the Clipboard.
4. Position the cursor in the new table at the location where you want to insert the columns.

5. Click the **Paste** button on the toolbar. The columns are inserted at the new location and remain highlighted until you click elsewhere.
6. Create an update query to add the data to the table to which you copied the columns.

[How?](#)

Caution You cannot undo the action of executing an Update query. As a precaution, back up your data before executing the query.

[Background information](#)

You can change the contents of multiple rows in one operation by using an Update query. For example, in a `titles` table you can use an Update query to add 10% to the price of all books for a particular publisher.

When you create an Update query, you specify:

- The table to update.
- The columns whose contents you want to update.
- The value or expression to use to update the individual columns.
- Search conditions to define the rows you want to update.

For example, the following query updates the `titles` table by adding 10% to the price of all titles for one publisher:

```
UPDATE titles
SET price = price * 1.1
WHERE (pub_id = '0766')
```

[Create an Update query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the table, view, or function you want to use to update, and then click **Close**.
4. On the **Query** menu click **Append Query**.

Note If more than one table is displayed in the Grid pane when you start the Update query, the Query Designer displays the **Update Table** dialog box to prompt you for the name of the table to update.

5. Define the data columns to update by adding them to the query. Columns will be updated only if you add them to the query.
6. In the **New Value** column of the Grid pane, enter the update value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

Caution The Query Designer cannot check that a value fits within the length of the column you are updating. If you provide a value that is too long, it might be truncated without warning. For example, if a name column is 20 characters long but you specify an update value of 25 characters, the last 5 characters might be truncated.

7. Define the rows to update by entering search conditions in the **Criteria** column.

If you do not specify a search condition, all rows will be updated.

Note When you add a column to the Grid pane for use in a search condition, the Query Designer also adds it to the list of columns to be updated. If you want to use a column for a search condition but not update it, clear the check box next to the column name in the rectangle representing the table, view, or function . ().

When you execute an Update query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were changed.

7. Run the update query.

[How?](#)

When you have finished designing your query, you can run it.

1. If you want to test whether the syntax of the query you are creating is correct, you can verify the query.

[How?](#)

To avoid problems, you can check the query you have built to ensure its syntax is correct. This option is especially useful when you enter statements in the SQL pane.

Note A statement can be valid, and therefore be verified successfully, even if it cannot be represented in the Diagram and Grid panes.

Note SQL Verification can detect some, but not all SQL errors. If a query contains an error not detected during SQL verification, the database will detect the error when you run the query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Click **Verify SQL Syntax** .
2. Click **Run** .

Notes

- If you are creating a Select query, the results of the query appear in datasheet view.
 - If you are creating an Update, Append, Append Values, Delete, or Make-Table query, the Query Designer displays a message indicating how many rows were affected by the query.
 - To stop a query that is taking too long to finish, press CTRL+BREAK.
8. Return to the database diagram and delete the columns from the original table.

Save your work in Table Designer (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can save all the changes you made to a table in the Table Designer.

1. In the Table Designer, click **Save** .
2. If you made changes to the table, the **Save** dialog box appears and displays a list of changes that will be saved in the database when you save your diagram.

Note The list of tables may contain tables other than the table you modified. For example, if you change the data type of a column that participates in a relationship with another table, both tables will be included in this list.

3. Click **Yes** to update the database.

Rename a column (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The name of a column in a table in your database diagram shows the name of the column as it is stored in the database. You can rename a column directly in your table in the database diagram and the database will be updated with the new name when you save the table or database diagram.

Column names are stored in the case (uppercase or lowercase) in which they appear in your database diagram.

Caution Renaming a column may affect triggers, stored procedures, constraints, views, and user-defined functions. Consider renaming a column before you create these other objects.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose column you want to rename.
3. If only the table name is shown in the database diagram, right-click the table and select **Column Properties**, **Column Names**, **Keys**, or **Custom View** from the shortcut menu.
4. In the cell that shows the column name you want to change, type a new column name.

The column is renamed in your table or database diagram as soon as you exit the cell that shows the column name. The column is renamed in your database when you save the table or database diagram.

Rename a table (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you rename a table, the table name is automatically updated in every database diagram in which the table appears. It is updated in the database when you save the table or database diagram. You must have the appropriate permission to rename a table. For more information on permissions, see the Microsoft SQL Server documentation.

Caution Think carefully before you rename a table. If existing queries, views, user-defined functions, stored procedures, or programs refer to that table, the name modification will make these objects invalid.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table you want to rename.
3. Right-click the table, and choose **Properties** from the shortcut menu.
4. Choose the **Tables** tab.
5. In the **Table name** box, type a new name. Be sure to choose a name that does not duplicate one in the **Selected table** list.
To cancel this action, press the ESC key before leaving this field.
6. Close the **Properties** dialog box.

The table is renamed in the database diagram as soon as you exit the **Table name** box.

[Show All](#)

Resize property columns (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can manually change the width of a property column in a database table by resizing it in column properties view. For example, you can widen the Datatype column if it contains user-defined data type names that are too long to see in their entirety. You can also automatically resize a property column to fit its widest entry.



[Manually change the width of a property column](#)

1. Click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose property columns you want to resize.
3. Position the pointer along the right border of the column header for the column that you want to resize. The cursor changes from an arrow to a splitter pointer ↔.
4. Drag the column border to the left to reduce its width or to the right to increase its width.
5. Release the mouse button.



[Resize a property column to fit the widest entry](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose columns you want to resize.
3. If necessary, right-click the table, and then click **Column Properties** or **Custom View**.
4. Position the pointer along the right border of the column header for the column that you want to resize. The cursor changes from an arrow to a splitter pointer ↔.

5. Double-click the splitter pointer.

Note You can also resize property columns in Custom view for any columns you have chosen to display in this view.

[Show All](#)

About tables (ADP)

[Overview of tables](#)

A database consists of one or more tables. A table is a collection of data, arranged in rows and columns. For example, you might have a table for author information called authors. Each column would contain a certain type of information, such as the author's last name. Each row would contain all the information about a specific author: first name, last name, address, and so on.

In a database, you might have a number of tables, each devoted to a specific topic. For example, the pubs database might contain tables for authors, titles, and so on. Using a separate table for each topic can eliminate duplicate data, make data storage more efficient, and reduce data-entry errors.

Tables are the basic building blocks of database diagrams. In a database diagram, each table is laid out in a matrix so that you can see all the properties defined for every column in your database table.

[Tables in a database diagram](#)

Each table in a database diagram has three distinct features: a title bar, a row selector, and a set of property columns.

- **Title Bar** The title bar shows the name of the table. If another user owns the table, then that user's name appears in parentheses at the end of the table name.



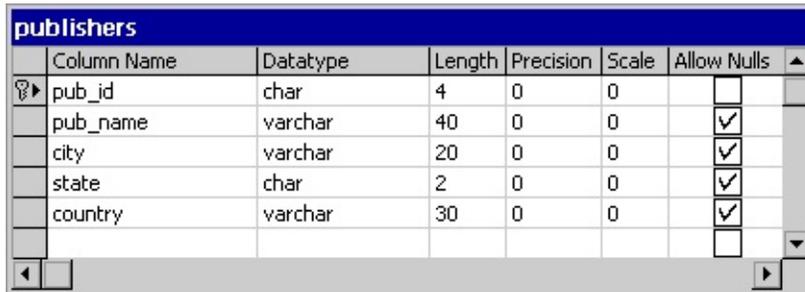
If you have modified a table and have not yet saved it, an asterisk (*) appears at the end of the table name to indicate unsaved changes.

 or 

- **Row Selector** You can click the row selector  to select a database

column in the table. The row selector displays a key symbol  if the column is in the table's primary key.

- **Property Columns** The set of property columns is visible only in certain views of your table. You can view a table in any of five different views to help you manage the size and layout of your diagram.



	Column Name	Datatype	Length	Precision	Scale	Allow Nulls	
	pub_id	char	4	0	0	<input type="checkbox"/>	
	pub_name	varchar	40	0	0	<input checked="" type="checkbox"/>	
	city	varchar	20	0	0	<input checked="" type="checkbox"/>	
	state	char	2	0	0	<input checked="" type="checkbox"/>	
	country	varchar	30	0	0	<input checked="" type="checkbox"/>	
						<input type="checkbox"/>	

Before you begin defining the columns for a table, determine what type of data the table will hold and how that table relates to the other tables in your database.

[Designing tables](#)

To determine the structure of a new table, you need to decide:

- What type of data the table will contain.
- What columns you need in the table and the data type (and length, if required) for each column.
- Which columns should accept null values.
- Whether to use constraints and if so, where.
- What types of indexes you need, where you need them, and which columns should be the primary key and foreign key.

After you decide on the structure of your table, you can create the table and define its columns in your database diagram or with the Table Designer. You can also alter the table's appearance in your database diagram so that the information you need is visible when you need it. When you save your table or the database diagram, the table is created in your database.

If you know exactly what you want in a table, it is often most efficient to define everything you need at the beginning, including the table's data restrictions and additional properties. However, in many cases, you will do best to first create a

basic table and save it so it is created in your database. You can then add some test data to the table and experiment with the table in the database diagram to fine-tune its design.

The Database Designer lets you try out different designs by working with tables in your database diagram. Through experimentation, you can determine what types of data are frequently entered and queried and then redesign your table accordingly.

When you change a table's design in a database diagram or in Table Designer, any data that is stored in the table is preserved to the extent possible. When you are satisfied with your basic design, you can add constraints, indexes, and any additional columns that you require.

Replace a global default value with a default value (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. If necessary, right-click the table and select **Column Properties**, **Column Names**, **Keys**, or **Custom View** from the shortcut menu.
3. Assign a new default value to the column you want to change.
4. A message prompts you to permanently unbind the existing default in order for the new default to be applied. Choose **OK**.

[Show All](#)

Troubleshoot keys, relationships, and indexes (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[I'm having problems creating or modifying a primary key.](#)

Keep in mind the following guidelines when creating or modifying a primary key:

- All columns that participate in a table's primary key must contain unique values other than [Null](#). To add the column to a primary key, first clear the **Allow Nulls** check box on the **Column Property** dialog box.
- A primary key must contain at least one column and contain unique data values.
- You cannot create a primary key using columns with a **text**, **ntext**, **image**, or **bit** data type.
- The maximum number of columns you can use to create a primary key is 16.
- The sum of the lengths of all columns in a primary key cannot exceed 900 bytes.

[I'm having problems creating or modifying a database relationship between table columns.](#)

Keep in mind the following guidelines when creating or modifying a database relationship and corresponding columns:

- A database relationship must relate at least one column from the foreign-key table to at least one column in the primary key table.
- At least one of the two tables must have a primary key or unique constraint defined and the columns on the primary key side of a database relationship must participate in either a primary key or a unique constraint.
- Both sides of the relationship must have the same number of columns; the

maximum number is 16 columns for each side of the relationship.

- The **Data Type**, **Length**, **Precision**, **Scale**, and **Collation** properties of each matching columns between the tables must be the same.
- Deleting columns that participate in relationships will delete the relationships associated with them.
- You cannot change a table's primary key or unique constraint while other foreign key relationships are still in place. To change a table's primary key or unique constraint, first delete all existing relationships for the table.
- A computed column cannot participate in relationship. Adding a formula to a column will delete all relationships it participates in. Either clear the column's **Formula** property or remove the column from the index or relationship.
- Removing a formula from a column may leave it with a data type that is disallowed in relationships.
- If you change the data type of a column that is related to columns in other tables, then the data type of the related columns must also be changed to preserve referential integrity. When you save the table or database diagram, Access displays the **Datatype Change Required** dialog box so you can automatically change the data type of the related columns.

[I can't see the join lines in my relationships when connected to SQL Server 6.5.](#)

The Microsoft [OLE DB Provider](#) for SQL Server 6.5 (SQL Service Pack 5) is implemented as a [dynamic link library \(DLL\)](#) on the client machine and a set of [stored procedures](#) on the server. Unless both parts are installed, the OLE DB provider will not function correctly. Also, if you install a newer version of `Sqloledb.dll`, but do not install `Instcat.sql` on each Microsoft SQL Server machine running OLE DB, some OLE DB functions will return incorrect results.

To resolve this, you should run `Instcat.sql` to install the stored procedures on the Microsoft SQL Server you wish to access with OLE DB.

For more information on installing SQL Server, see the SQL Server documentation.

[I'm having problems creating, modifying, or deleting an index.](#)

Keep in mind the following guidelines when creating, modifying, or deleting an

index or a column that participates in an index:

- An index must contain at least one column. If you are creating a unique index, that column must contain unique data values.
- You cannot create indexes of columns with a **text**, **ntext**, **image**, or **bit** data type.
- The maximum number of columns you can use to create an index is 16.
- The sum of the lengths of all columns in an index cannot exceed 900 bytes.
- Be careful deleting a column or changing its data type. If the column participates in an index, you may end up deleting the index unintentionally.
- You can only create one clustered index on a table. A clustered index is a special type of index that reorders the way records in the table are physically stored. If you want to create two or more indexes on the same table, the additional indexes must be non-clustered indexes.
- In SQL Server 7.0 and 6.5 databases, a computed column cannot participate in an index. Either clear the column's **Formula** property or remove the column from the index.
- Columns you modify may contribute to one or more indexes. If you modify the column and it becomes a computed column, it cannot contribute to indexes anymore.

[Show All](#)

About constraints (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Constraints are business logic that your database server enforces for you. They limit the possible values that users can enter into specified columns, enforcing referential integrity. When you create constraints in the Database Designer, they conform to the ANSI standard for creating and altering tables.

 [Constraints and referential integrity](#)

When you update a table using the Query Designer, the query respects any constraint defined in the database for the columns and tables that you are updating. For example, if a table is defined with a unique constraint, you cannot execute an Update or Append query that would write a non-unique row to the table. Similarly, if you are updating a column that has a constraint limiting its values to a certain range, the query will result in an error if you provide an update value outside the specified range.

Queries also respect referential integrity defined between tables. For example, you cannot add rows to a table in the "many" side of a one-to-many relationship if the corresponding row in the "one" side does not exist.

 [Types of constraints](#)

 [Primary key constraints](#)

A primary key constraint ensures no duplicate values are entered in particular columns and that NULL values are not entered in those columns. You can use primary key constraints to enforce uniqueness as well as referential integrity. For example, the au_id column uniquely identifies each author stored in the authors table.

You create primary key constraints directly in a database diagram.

[Foreign key constraints](#)

A foreign key constraint works in conjunction with primary key or unique constraints to enforce referential integrity among specified tables. For example, you can place a foreign key constraint on the `title_id` column in the `publishers` table to ensure that a value entered in that column matches an existing value in the `title_id` column of the `titles` table.

In database diagrams, a foreign key constraint is automatically placed on specified columns when you create a relationship to a table from another table to which a primary key or unique constraint is attached.

[Check constraints](#)

A check constraint specifies the data values or formats that are acceptable in one or more columns in a table. For example, you can require the `zip` column of the `authors` table to allow only five-digit numeric entries.

You can define many check constraints for a table. You use the **Constraints** property page to create, modify, or delete each check constraint.

[Check constraint expressions](#)

[Overview of check constraint expressions](#)

When you attach a check constraint to a table or column, you must include an SQL expression.

You can create a simple constraint expression to check data for a simple condition; or you can create a complex expression, using Boolean operators, to check data for several conditions. For example, suppose the `authors` table has a `zip` column where a 5-digit character string is required. This sample constraint expression guarantees that only 5-digit numbers are allowed:

```
zip LIKE '[0-9][0-9][0-9][0-9][0-9]'
```

Or suppose the `sales` table has a column called `qty` which requires a value greater than 0. This sample constraint guarantees that only positive values are

allowed:

```
qty > 0
```

Or suppose the `orders` table limits the type of credit cards accepted for all credit card orders. This sample constraint guarantees that if the order is placed on a credit card, then only Visa, MasterCard, or American Express is accepted:

```
NOT (payment_method = 'credit card') OR  
(card_type IN ('VISA', 'MASTERCARD', 'AMERICAN EXPRESS'))
```



[Defining a constraint expression](#)

In the **Check Constraints** tab of the property page, type an expression in the **Constraint expression** box using the following syntax:

```
{constant | column_name | function | (subquery)}  
[ {operator | AND | OR | NOT }  
{constant | column_name | function | (subquery)}...]
```

The SQL syntax is made up of the following parameters:

Parameter	Description
<i>constant</i>	A literal value, such as numeric or character data. Character data must be enclosed within single quotation marks ('').
<i>column_name</i>	Specifies a column.
<i>function</i>	A built-in function. For details about functions, see the SQL Server documentation.
<i>operator</i>	An arithmetic, bitwise, comparison, or string operator. For details about operators, see the SQL Server documentation. Use in Boolean expressions to connect two expressions. Results are returned when both expressions are true.
AND	When AND and OR are both used in a statement, AND is processed first. You can change the order of execution by using parentheses. Use in Boolean expressions to connect two or more conditions.

Results are returned when either condition is true.

OR When AND and OR are both used in a statement, OR is evaluated after AND. You can change the order of execution by using parentheses.

Negates any Boolean expression (which can include keywords, such as LIKE, NULL, BETWEEN, IN, and EXISTS).

NOT When more than one logical operator is used in a statement, NOT is processed first. You can change the order of execution by using parentheses.

[Unique constraints](#)

A unique constraint ensures no duplicate values are entered into specified columns that are not a table's primary key. For example, in the employee table in which the emp_id column is the primary key, you can define a unique constraint that requires entries in the Social Security number (ssn) column to be unique within the table.

In database diagrams, you use the **Indexes/Keys** property page to create, modify, or delete unique constraints.

[Default constraints](#)

A default constraint enables you to define the value that will be supplied for a column whenever a user fails to enter a value. For example, in a table with a column called payterms, you can instruct your database server to enter "???" or "fill in later" if the user leaves it blank.

In database diagrams, you define a default constraint as a property of a column in your table. You define this type of constraint for a column by specifying a default value inside a table in column properties view. Be sure to specify the constraint with the correct delimiters. For example, strings must be surrounded with single quotes.

[Show All](#)

Create a constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Define a primary key constraint](#)

Define a primary key to enforce uniqueness for values entered in specified columns that do not allow nulls. If you define a primary key for a table in your database, you can relate that table to other tables, thus reducing the need for redundant data. A table can have only one primary key.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector for the database column you want to define as the primary key. If you want to select multiple columns, hold down the CTRL key while you click the row selectors for the other columns.
3. Right-click the row selector for the column and select **Primary Key**. A primary key index, named "PK_" followed by the table name, is automatically created; you can find it on the **Indexes/Keys** tab of the **Properties** page.

Warning If you want to redefine the primary key, any relationships to the existing primary key must be deleted before the new primary key can be created. A message will warn you that existing relationships will be automatically deleted as part of this process.

A primary key column is identified by a primary key symbol  in its row selector.

If a primary key consists of more than one column, duplicate values are allowed in one column, but each combination of values from all the columns in the primary key must be unique.

If you define a compound key, the order of columns in the primary key matches

the order of columns as shown in the table in your database diagram. However, you can change the order of columns after the primary key is created. In the **Column name** grid, remove the columns from the primary key. Then add the columns back in the order you want. To remove a column from the key, simply remove the column name from the Column name list.

[Attach a new check constraint to a table or column](#)

Attach a check constraint to a table to specify the data values that are acceptable in one or more columns.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table that will contain the constraint, then select **Constraints** from the shortcut menu.
3. Choose **New**. The **Selected constraint** box displays the system-assigned name of the new constraint. System-assigned names begin with "CK_" followed by the table name.
4. In the **Constraint expression** box, type the SQL expressions for the check constraint. For example, to limit the entries in the state column of the authors table to New York, type:

```
state = 'NY'
```

Or, to require entries in the zip column to be 5 digits, type:

```
zip LIKE '[0-9][0-9][0-9][0-9][0-9]'
```

Note Make sure to enclose any non-numeric constraint values in single quotation marks (').

5. If you want to give the constraint a different name, type the name in the **Constraint name** box.
6. Use the check boxes to control when the constraint is enforced:
 - To test the constraint on existing data before creating the constraint, check **Check existing data on creation**.
 - To enforce the constraint whenever a replication operation occurs on

this table, check **Enforce constraint for replication**.

- To enforce the constraint whenever a row of this table is inserted or updated, check **Enforce constraint for INSERTs and UPDATEs**.

[Check existing data when creating a check constraint](#)

When you create a check constraint, you can set an option to apply it either to new data only or to existing data as well. The option of applying the constraint to new data only is useful when you know that the existing data already meets the new check constraint, or when a business rule requires the constraint to be enforced only from this point forward.

For example, you may have required zip codes to be limited to five digits in the past, but now want new data to allow nine-digit zip codes. Old data with five-digit zip codes will coexist with new data that contains nine-digit zip codes.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.

-or-

In your database diagram, right-click the table containing the constraint, then select **Constraints** from the shortcut menu.

2. Select the constraint from the **Selected constraint** list.
3. Select the **Check existing data on creation** check box. This option is selected by default.

The check constraint will be applied when you save the database diagram. If any constraint violations are encountered during the save process, the table cannot be saved.

[Create a unique constraint](#)

Create a unique constraint to ensure no duplicate values are entered in specific columns that do not participate in a primary key. While both a unique constraint and a primary key enforce uniqueness, you should attach a unique constraint instead of a primary key constraint to a table if:

- You want to enforce uniqueness in a column or combination of columns. You can attach multiple unique constraints to a table, whereas you can attach only one primary key constraint to a table.
- You want to enforce uniqueness in a column that allows null values. You can attach unique constraints to columns that allow null values, whereas you can attach primary key constraints only to columns that do not allow null values. When you attach a unique constraint to a column allowing null values, you ensure that at most one row will have a null value in the constrained column.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table that will contain the constraint, then select **Indexes/Keys** from the shortcut menu.
3. Choose **New**. A system-assigned name appears in the **Index name** box.
4. Under **Column name**, expand the list of columns and select the column that you want to attach the constraint to. To attach the constraint to multiple columns, select the additional columns in subsequent rows.
5. Select the **Create UNIQUE** check box.
6. Select the **Constraint** option.

The unique constraint is created in the database when you save the database diagram.

You can control the sort order of key values and the action taken when duplicate keys exist. To do this, you should create a unique index instead of a unique constraint.

[Show All](#)

Create constraint validation text (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When your query encounters a [constraint](#) violation, by default Microsoft Access displays a message from Microsoft SQL Server. If your [Access project](#) is connected to a SQL Server 2000 database, you can create and show a more user-friendly message using the **Validation Text** property.

1. In the [Database window](#), click **Tables**  under **Objects**, select the table you want, and then click **Open** on the database window toolbar.
2. Right click on the Design grid, and then select **Constraints**.
3. In the **selected constraint** list box, select the constraint you want, or click **New** to create a new constraint.
4. Type the user message in the **Validation Text** text box as the following examples show.

If the constraint expression is:

Type the validation text:

MinJobLevel > 10

"The job level is too low. You must enter a number greater than 10."

HouseholdName Is Not Null

"Please enter a household name."

Quantity >= 0

"You must enter a positive number."

5. Close the **Properties** sheet to save your changes.

Note The **Validation Text** property is ignored in an unbound form that fills in control values directly from [ADO](#). In this case, you must trap the error and display the validation text in your code.

[Show All](#)

Delete a constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Delete a primary key constraint](#)

Delete a primary key constraint when you want to remove the requirement for uniqueness for the values entered in a column or a combination of columns.

 [Delete a primary key constraint by creating a new primary key](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the primary key columns for the table whose primary key constraint you want to delete. A primary key column is identified by a primary key symbol  in its row selector.
3. Right-click the row selector for the column and select **Primary Key**.

 [Delete a primary key constraint](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table whose primary key constraint you want to delete.
3. Right-click the table and select **Indexes/Keys**.
4. Select the primary key index from the **Selected index** list.
5. Choose **Delete**.

Caution Choosing **Delete** will result in an action that cannot be undone without losing all other changes made to the database diagram. To undo this action, close this database diagram and all other open database diagrams without saving the changes.

The constraint is deleted from the database when you save the database diagram.

[Delete a foreign key constraint](#)

Delete a foreign key constraint when you want to remove the requirement to enforce referential integrity.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, delete the relationship line that represents the foreign key constraint you want to delete.

[How?](#)

Delete a relationship when you no longer want to relate columns in two related tables. When you redesign tables, it is often necessary to delete relationships and then recreate them after your new design is complete. For example, if you decide to normalize a database and store all address data in one table, you would delete all the relationships to address columns between existing tables, create a new table containing the address columns, and then create relationships from the new address table to every table that requires an address.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the line that represents the relationship that you want to delete from the diagram.
3. Right-click the relationship line and choose **Delete Relationship from Database** from the shortcut menu.
4. A message box prompts you to confirm the deletion. Choose **Yes**.

Note When you delete a relationship, the relationship line is removed from every diagram in which it appears. It is deleted from the database when you save the database diagram or when you save either of the tables that it related.

Note Deleting a relationship from a database diagram marks the related

tables as modified in all database diagrams in which they appear.

[Delete a check constraint](#)

Delete a check constraint when you want to remove the limitations on data values that are accepted in the column or columns included in the constraint expression.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Constraints** from the shortcut menu.
3. Select the constraint from the **Selected constraint** list.
4. Choose **Delete**.

Caution Choosing **Delete** will result in an action that cannot be undone without losing all other changes made to the database diagram. To undo this action, close this database diagram and all other open database diagrams without saving the changes.

The constraint is deleted from the database when you save the database diagram.

[Delete a unique constraint](#)

Delete a unique constraint when you want to remove the requirement for uniqueness for values entered in the column or combination of columns included in the constraint expression.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constrained column or columns, then select **Indexes/Keys** from the shortcut menu.
3. Select the unique constraint from the **Selected index** list.
4. Choose **Delete**.

Caution Choosing **Delete** will result in an action that cannot be undone without losing all other changes made to the database diagram. To undo this

action, close this database diagram or table design window and all other open database diagrams and table design windows without saving the changes.

The constraint is deleted from the database when you save the database diagram.

[Show All](#)

Disable a constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Disable a foreign key constraint with INSERT or UPDATE Statements](#)

Select the option to disable a foreign key constraint during INSERT and UPDATE transactions if you know that new data will violate the constraint or if the constraint applies only to the data already in the database.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the foreign key, then select **Relationships** from the shortcut menu.
3. Select the relationship from the **Selected relationship** list.
4. Clear the **Enforce relationship for INSERTs and UPDATEs** check box.

After you add or modify data, you should select this option if you want to ensure the constraint applies to subsequent data modifications.

Note If you plan to use triggers to implement database operations, you must disable foreign key constraints in order for the trigger to run.

[Disable a check constraint with INSERT or UPDATE Statements](#)

[Background information](#)

You can disable a check constraint when data is added to, updated in, or deleted from a table. Disabling a constraint enables you to perform the following transactions:

- Add a new row of data to a table (using the INSERT statement) where the existing rows were required to meet specific business rules that no longer apply. For example, you may have required postal codes to be limited to

five digits in the past, but now want new data to allow nine-digit postal codes. Old data with five-digit postal codes will coexist with new data that contains nine-digit postal codes.

- Modify existing rows (using the UPDATE statement) where the existing rows were required to meet specific business rules that no longer apply. For example, you may want to update all existing five-digit postal codes to nine-digit postal codes.

Select the option to disable a check constraint during INSERT and UPDATE transactions if you know that new data will violate the constraint, or if the constraint applies only to the data already in the database.

[Disable a check constraint with INSERT and UPDATE statements](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Properties** from the shortcut menu.
3. Choose the **Check Constraints** tab.
4. Select the constraint from the **Selected constraint** list.
5. Clear the **Enforce constraint for INSERTs and UPDATEs** check box.

You can select this option after you add or modify data to guarantee that the constraint applies to subsequent data modifications.

[Disable a foreign key constraint for replication](#)

Microsoft SQL Server supports replication. Select the option to disable a foreign key constraint during replication if the constraint is specific to the source database and may unnecessarily prevent new data from being entered into the destination database.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the foreign key, then select **Properties** from the shortcut menu.
3. Choose the **Relationships** tab.

4. Select the relationship from the **Selected relationship** list.
5. Clear the **Enforce relationship for replication** check box.

[Disable a check constraint for replication](#)

You can disable a check constraint when your table is replicated in another database. When you replicate a table, the table definition and data are copied from the source database to a destination database. These two databases are usually (but not necessarily) on separate servers. If the check constraints are specific to the source database, they may unnecessarily prevent new data from being entered in the destination database. When you replicate a database at a remote site, you should not reapply check constraints because:

- The integrity of the data was checked when it was entered into the original database.
- The replication will fail if data violates the check constraints.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Properties** from the shortcut menu.
3. Choose the **Check Constraints** tab.
4. Select the constraint from the **Selected constraint** list.
5. Clear the **Enforce constraint for replication** check box.

[Show All](#)

Modify or rename a constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Modify a primary key constraint](#)

Modify a primary key if you want to change the column order, index name, clustered option, or fill factor.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table whose primary key you want to modify, and choose **Indexes/Keys** from the shortcut menu.
3. Select the primary key index from the **Selected index** list.
4. Type a new name in the **Index name** box. Make sure that your new name does not duplicate a name in the **Selected index** list.

The primary key is updated in the database when you save your database diagram.

[Modify a foreign key constraint](#)

Modify the foreign key side of a relationship if you want to change which columns are related to columns in the primary key table.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the relationship corresponding to the foreign key you want to modify, and then choose **Properties** from the shortcut menu.
3. Choose the **Relationships** tab.
4. Select the relationship from the **Selected relationship** list.
5. In the **Foreign key table** column, expand the list in the first row.

6. Select a different table column from the list. The foreign key column must match the data type and size of the primary key column, with these exceptions:
 - A char column or sysname column can relate to a varchar column.
 - A binary column can relate to a varbinary column.
 - A user-defined data type can relate to its base type.

Any changes you make to the relationship's properties take effect as soon as you move outside the grid in the property pages. The constraint is updated in the database when you save your database diagram.

Note Modifying a relationship in a database diagram marks both of the related tables as modified. Consequently, each table will also be marked as modified in any other database diagrams in which it appears.

[Rename a check constraint](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Constraints** from the shortcut menu.
3. Select the constraint you want to change from the **Selected constraint** list.
4. Type a new name in the **Constraint name** box.

The constraint name is updated in the database when you save your database diagram.

[Modify a unique constraint](#)

Modify a unique constraint when you want to change the columns that the constraint is attached to, change the constraint name, or set additional properties for the constraint.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Indexes/Keys** from the shortcut menu.

3. Select the constraint you want to change from the **Selected index** list.
4. Type a new name in the **Index name** box. Make sure that your new name does not duplicate a name in the **Selected index** list.

The constraint is updated in the database when you save your database diagram.

View foreign key constraint attributes (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

View the foreign key attributes of a relationship if you want to see which columns participate in the foreign key side of a relationship. If the foreign key columns are related to a primary key, the primary key columns are identified in your database diagram by a primary key symbol in the row selector .

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the relationship line corresponding to the foreign key, then choose **Properties** from the shortcut menu.
3. Choose the **Relationships** tab.
4. Make sure the relationship is selected in the **Selected relationship** list.
5. The **Foreign key table** column displays the name of each column that participates in the foreign key side of the relationship.

About keys (ADP)

There are two kinds of keys. A primary key is a set of columns from a table that are guaranteed to have unique values for each row of that table. A primary key is also called a primary key constraint, because it effectively constrains the values you can add to the table: it prevents you from adding a row to the table whose primary key columns are all equal to the corresponding values of some other row in that table.

A foreign key is a correspondence between a set of columns in one table and the set of primary key columns in some other table. When discussing foreign keys, the two participating tables are sometimes called the foreign-key table and the primary-key table. A foreign key is also called a foreign key constraint because it constrains table rows: it ensures that any row you add to the foreign-key table has a corresponding row in the primary-key table. That is, it requires that any row added to the foreign-key table have values in the foreign-key column that correspond to the respective values of the primary key columns for some row in the primary-key table.

Delete an index (ADP)

Indexes can slow INSERT, UPDATE, and DELETE performance. If you find that an index hinders overall performance or you no longer need it, you can delete it.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table whose index you want to delete, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab.
4. From the **Column Name** list, select the index you want to delete.
5. Choose **Delete**.

Caution Choosing **Delete** will result in an action that cannot be undone without losing all other changes made to the database diagram. To undo this action, close this database diagram and all other open database diagrams and Table Designer windows without saving the changes.

The index is deleted from the database when you save your database diagram.

Rename an index (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

New indexes are automatically given system-defined names based on the database table name. If you create multiple indexes on a table, the index names are appended with "_1", "_2", and so on. You can rename an index as long as index names are unique within the table.

Note When you create a primary key or unique constraint for a table, an index with the same name as the constraint is automatically created for the table. Because index names must be unique for a table, you cannot create or rename an index to have the same name as the primary key or unique constraint for the table.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table whose index you want to rename, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab.
4. Select the index from the **Column name** list.
5. Type a new name in the **Index name** box. Make sure that it does not duplicate a name in the **Column name** list.

The index is renamed in your database diagram when you exit the **Index name** box. It is renamed in the database when you save the database diagram.

[Show All](#)

Specify a fill factor for an index (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

In Microsoft SQL Server databases, you can identify a fill factor to specify how full each index page can be. The amount of empty space on an index page is important because when an index page fills up, the system must take time to split it to make room for new rows.

It is seldom necessary to specify a fill factor when you create an index. The option is provided for fine-tuning performance. It is useful when you are creating a new index on a table with existing data, and particularly when you can accurately predict future changes in that data.

[Specify a fill factor for an index on a table](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, right-click the table containing the index for which you want to specify a fill factor, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab.
4. Select the index from the **Column name** list.
5. In the **Fill factor** box, type a percentage from 0 to 100.

[Specify a fill factor for an index on a view \(Microsoft SQL Server 2000 only\)](#)

1. In the Database window, click **Queries**  under **Objects**, click the view you want to open, and then click **Design** on the database window toolbar.
2. Right-click in the Query Designer, and choose **Manage Indexes** from the shortcut menu.

3. Select the index from the **Column name** list.
4. In the **Fill factor** box, type a percentage from 0 to 100.

For more information about fill factors and their uses, see the SQL Server documentation.

Update an index (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Your index is automatically saved in the database when you save your table or database diagram. The index is available for modification as soon as you create it.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, right-click the table and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab.
4. To update the definition, select the option you want to update and change its value or setting. For more information on each property, click **Help**.

Your changes are saved to the database when you save the database diagram.

Note that you can also manage indexes on some views.

[Show All](#)

About indexes (ADP)

[What is an index?](#)

You can use an index to gain fast access to specific information in a database table. An index is a structure that orders the values of one or more columns in a database table, for example the last name (lname) column of the employee table. If you were looking for a specific employee by his or her last name, the index would help you get that information faster than if you had to search all the rows in the table.

The index provides pointers to the data values stored in specified columns of the table, and then orders those pointers according to the sort order you specify. The database uses the index much as you use an index in a book: it searches the index to find a particular value and then follows the pointer to the row containing that value.

In database diagrams, you can create, edit, or delete each type of index in the **Indexes/Keys** property page for a selected table. An index is saved in the database when you save the table that it is attached to, or when you save the database diagram in which that table appears.

As a general rule, you should create an index on a table only if the data in the indexed columns will be queried frequently. Indexes take up disk space and slow the adding, deleting, and updating of rows. In most situations, the speed advantages of indexes for data retrieval greatly outweigh these disadvantages. However, if your application updates data very frequently or if you have disk space constraints, you might want to limit the number of indexes.

Before creating an index, you must determine what columns to use and what type of index to create.

[Index columns](#)

You can create indexes based on a single column or on multiple columns in a database table. Multiple-column indexes enable you to distinguish between rows

in which one column may have the same value.

Indexes are also helpful if you often search or sort by two or more columns at a time. For example, if you often set criteria for last name and first name columns in the same query, it makes sense to create a multiple-column index on those two columns.

To determine the usefulness of an index:

- Examine the WHERE and JOIN clauses of your queries. Each column included in either clause is a possible candidate for an index.
- Experiment with the new index to examine its effect on the performance of running queries.
- Consider the number of indexes already created on your table. It is best to avoid a large number of indexes on a single table.
- Examine the definitions of the indexes already created on your table. It is best to avoid overlapping indexes that contain shared columns.
- Examine the number of unique data values in a column and compare that number with the number of rows in the table. The result is the selectivity of that column, which can help you decide if a column is a candidate for an index and, if so, what type of index.

[Types of indexes](#)

Depending on the functionality of your database, you can create three types of indexes - unique, primary key, and clustered - in Database Designer.

[Tip](#)

Although a unique index will help locate information, for the best performance results it is recommended that you use primary key or unique constraints instead.

[Unique Index](#)

A unique index is one in which no two rows are permitted to have the same index value.

Most databases prevent you from saving a table with a newly created unique index when there are duplicate key values in the existing data. Your database

may also prevent the addition of new data that would create duplicate key values in the table. For example, if you create a unique index on the employee's last name (lname) in the employee table, then no two employees can share the same last name.

[Primary Key Index](#)

A database table often has a column or combination of columns whose value uniquely identifies each row in the table. This column is called the primary key of the table.

Defining a primary key for a table in a database diagram automatically creates a primary key index that is a specific type of unique index. This index requires each value in the primary key to be unique. It also permits fast access to data when you use the primary key index in queries.

[Clustered Index](#)

In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the key values. A table can contain only one clustered index.

If an index is not clustered, the physical order of the rows in the table does not match the logical order of the key values. A clustered index usually provides faster access to data than does a nonclustered index.

[Show All](#)

Create an index (ADP)

You can use an index to speed access to data in a database table. You create an index by selecting one or more columns in a table that you want to be able to search on. You can use the index once you save the table.

[Create an index](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table you want to index, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab.
4. Choose **New**. The **Selected index** box displays the system-assigned name of the new index.
5. Under **Column name**, select the columns you want to index. You can select up to 16 columns. For optimal performance, select only one or two columns. For each column you select, you can indicate whether the index organizes its values in ascending or descending order.
6. Specify any other desired settings for the index. For more information on each property, click **Help**.

The index is created in the database when you save the database diagram.

[Create a unique index](#)

[Background information](#)

In SQL Server, you can create a unique index when uniqueness is a characteristic of the data itself, but the combination of indexed columns is not the same as the table's primary key. For example, if you plan to query frequently on the Social Security number (ssn) column in the employee table (where the primary key is emp_id), and you want to ensure Social Security numbers are unique, create a unique index on ssn. If the user enters the same Social Security number for more than one employee, the database displays an error and cannot save the table.

When you create or modify a unique index, you can set an option to ignore duplicate keys. If this option is set and you attempt to create duplicate keys by adding or updating data that affects multiple rows (with the INSERT or UPDATE statement), the row that causes the duplicates is not added or, in the case of an update, discarded.

For example, if you try to update "Smith" to "Jones" in a table where "Jones" already exists, you end up with one "Jones" and no "Smith" in the resulting table. The original "Smith" row is lost because an UPDATE statement is actually a DELETE followed by an INSERT. "Smith" was deleted and the attempt to insert an additional "Jones" failed. The whole transaction cannot be rolled back because the purpose of this option is to allow a transaction in spite of the presence of duplicates.

[Create a unique index](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table you want to index, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose **Indexes/Keys**. The **Selected index** list displays the system-assigned name of the new index.
4. Under **Column name**, select the columns you want to index. You can select up to 16 columns. For optimal performance, select only one or two columns per index. For each column you select, indicate whether the index arranges values of this column in ascending or descending order.
5. Select the **Create UNIQUE** check box.
6. Select the **Index** option.
7. Select the **Ignore duplicate key** option if you want to ignore new or updated data that would create a duplicate key in the index (with the INSERT or UPDATE statement).

The index is created in the database when you save the database diagram.

Note You cannot create a unique index on a single column if that column contains NULL in more than one row. Similarly, you cannot create a unique index on multiple columns if the combination of columns contains NULL in more than one row. These are treated as duplicate values for indexing purposes

[Create a clustered index](#)

[Background information](#)

In Microsoft SQL Server databases you can create a clustered index. In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the index key values. A table can contain only one clustered index. UPDATE and DELETE operations are often accelerated by clustered indexes because these operations require large amounts of data to be read. Creating or modifying a clustered index can be time-consuming, because it is during these operations that the table's rows are reorganized on disk.

Consider using a clustered index for:

- Columns that contain a limited number of unique values, such as a state column that contains only 50 unique state codes.
- Queries that return a range of values, using operators such as BETWEEN, >, >=, <, and <=.
- Queries that return large result sets.

[Create a clustered index](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table you want to index, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab, and then create a new index. For more information on each dialog box option, click **Help**.

To modify an existing index, select the index from the list.

4. Select the **Create as CLUSTERED** check box.

The index is created in the database when you save the database diagram.

[Show All](#)

About triggers (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).



[What is a trigger?](#)

A trigger is a special kind of stored procedure that goes into effect when you modify data in a specified table using one or more data modification operations: UPDATE, INSERT, or DELETE. Triggers can query other tables and can include complex SQL statements. They are primarily useful for enforcing complex business rules or requirements. For example, you could control whether to allow an order to be inserted based on a customer's current account status.

Triggers are also useful for enforcing referential integrity, which preserves the defined relationships between tables when you add, update, or delete the rows in those tables. However, the best way to enforce referential integrity is to define primary key and foreign key constraints in the related tables. If you use database diagrams, you can create a relationship between tables to automatically create a foreign key constraint.



[Advantages of using triggers](#)

Triggers are useful in these ways:

- Triggers are automatic: they are activated immediately after any modification to the table's data, such as a manual entry or an application action.
- Triggers can cascade changes through related tables in the database. For example, you can write a delete trigger on the `title_id` column of the `titles` table to cause a deletion of matching rows in other tables. The trigger uses the `title_id` column as a unique key to locate matching rows in the `titleauthor`, `sales`, and `roysched` tables.
- Triggers can enforce restrictions that are more complex than those defined with check constraints. Unlike check constraints, triggers can reference columns in other tables. For example, a trigger can roll back updates that

attempt to apply a discount (stored in the `discounts` table) to books (stored in the `titles` table) with a price of less than \$10.

Create a trigger (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

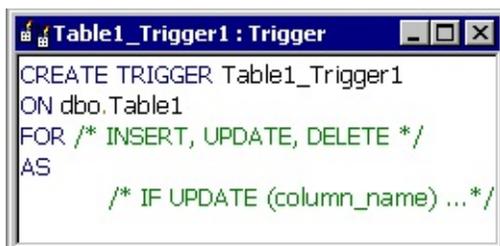
A trigger is a database object that you create by specifying:

- The current table.
- The data modification transactions that activate the trigger: adding new data (INSERT), updating existing data (UPDATE), or deleting existing data (DELETE).
- The actions that the trigger will take immediately following the transactions you specified.

You write triggers in Transact-SQL for MicrosoftSQL Server databases.

1. In the Database window, click **Tables**  under **Objects**.
2. Right-click the name of the table that you want to create a trigger on.
3. Choose **Triggers** from the shortcut menu.
4. In the **Triggers for Table: <table name>** dialog box, click **New**.

A new trigger is created in the SQL text source code editor with the default SQL statements already defined for you.



```
CREATE TRIGGER Table1_Trigger1
ON dbo.Table1
FOR /* INSERT, UPDATE, DELETE */
AS
    /* IF UPDATE (column_name) ... */
```

5. Modify the default trigger text as follows:

Replace	With
TABLE1_Trigger1	The name you want to assign to the trigger
dbo.TABLE1	The name of the table you want to attach the trigger to
/*Insert, Update,	The type of transactions that will activate this

```
Delete*/  
/*If Update  
(column_name) ...*/
```

trigger
The transactions that will take place

6. Write the remaining trigger text in SQL.

For syntax and examples of triggers, see the Microsoft SQL documentation.

Note Include both a RAISERROR statement and ROLLBACK transaction together in a trigger for situations when the trigger determines that a modification must be disallowed. This will ensure that the client record set stays in synch with the data at the server. If you include only the RAISERROR statement or ROLLBACK transaction, but not both, the client record set will not reflect the current state of the data at the server. The RAISERROR statement should be defined with a severity level greater than 10. For more information, see RAISERROR and ROLLBACK Transaction in the SQL Server documentation.

Delete a trigger (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you want to disable the actions defined in the trigger that are automatically carried out on your database immediately following the specified transactions, you can delete a trigger.

You might also want to delete any triggers that enforce referential integrity between related tables if you use database diagrams to design your database. Database diagrams use relationships instead of triggers for this purpose. Thus, if a trigger duplicates a relationship in a database diagram, you should delete either the trigger or the relationship.

1. In the Database window, click **Tables**  under **Objects**.
2. Right-click the table whose trigger you want to delete and choose **Triggers** from the shortcut menu.
3. In the **Triggers for Table: <table name>** dialog box, select the trigger you want to delete from the **Trigger Name** list, and then click **Delete**.
4. A message prompts you to confirm the deletion. Choose **Yes**.

Open a trigger (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can open a trigger to view or edit the text of an existing trigger that is stored in your database. Triggers are scripted in Transact-SQL for Microsoft SQL Server databases.

1. In the Database window, click **Tables**  under **Objects**.
2. Right-click the name of the table whose trigger you want to open and choose **Triggers** from the shortcut menu.
3. In the **Triggers for Table: <table name>** dialog box, select the trigger you want to open from the **Trigger Name** list, and then click **Edit**.

The trigger is opened in the SQL source code editor which you can use to edit the SQL statements. For more information about triggers and the CREATE TRIGGER SQL statement, see the Microsoft SQL Server documentation.

[Show All](#)

Rename a trigger (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you change the name of a trigger, the trigger name is updated in your database as soon as you save the trigger. However, you'll need to update any references to the previous name in other areas where you've referenced it, such as in code.

1. Open the trigger.

[How?](#)

You can open a trigger to view or edit the text of an existing trigger that is stored in your database. Triggers are scripted in Transact-SQL for Microsoft SQL Server databases.

1. In the Database window, click **Tables**  under **Objects**.
2. Right-click the name of the table whose trigger you want to open and choose **Triggers** from the shortcut menu.
3. In the **Triggers for Table: <table name>** dialog box, select the trigger you want to open from the **Trigger Name** list, and then click **Edit**.

The trigger is opened in the SQL source code editor which you can use to edit the SQL statements. For more information about triggers and the CREATE TRIGGER SQL statement, see the Microsoft SQL Server documentation.

2. In the SQL source code editor, replace the existing trigger name with the new name. (The trigger name appears after the Create Trigger statement.)
3. Click **Save** to save the new trigger.
4. Delete the old trigger (unless you want two identical triggers with different names).

[How?](#)

When you want to disable the actions defined in the trigger that are automatically carried out on your database immediately following the specified transactions, you can delete a trigger.

You might also want to delete any triggers that enforce referential integrity between related tables if you use database diagrams to design your database. Database diagrams use relationships instead of triggers for this purpose. Thus, if a trigger duplicates a relationship in a database diagram, you should delete either the trigger or the relationship.

1. In the Database window, click **Tables**  under **Objects**.
2. Right-click the table whose trigger you want to delete and choose **Triggers** from the shortcut menu.
3. In the **Triggers for Table: <table name>** dialog box, select the trigger you want to delete from the **Trigger Name** list, and then click **Delete**.
4. A message prompts you to confirm the deletion. Choose **Yes**.

The new trigger name is now in the database.

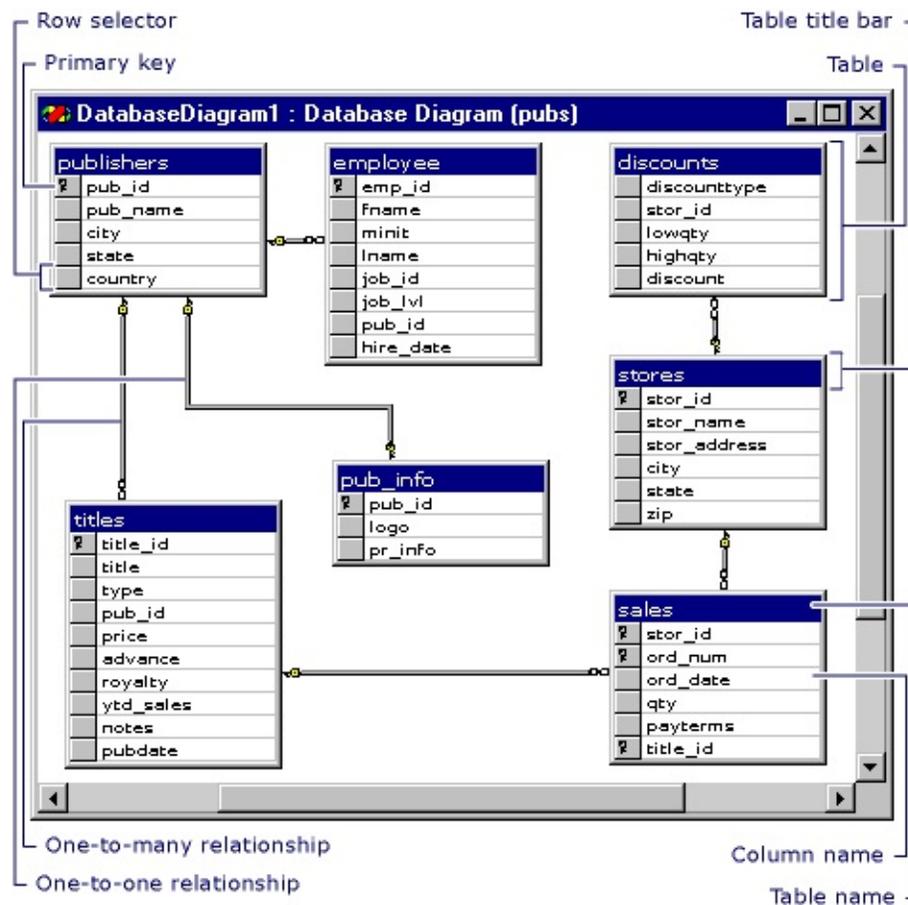
[Show All](#)

About Database Designer (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[What is the Database Designer?](#)

The Database Designer is a visual tool allowing you to design and visualize a database to which you are connected. When designing a database, you can use the Database Designer to create, edit, or delete tables, columns, keys, indexes, relationships, and constraints. To visualize a database, you can create one or more diagrams illustrating some or all of the tables, columns, keys, and relationships in it.



For any database, you can create as many database diagrams as you like; each

database table can appear on any number of diagrams. Thus, you can create different diagrams to visualize different portions of the database, or to accentuate different aspects of the design. For example, you can create a large diagram showing all tables and columns, and you can create a smaller diagram showing all tables without showing the columns.

Each database diagram you create is stored in the associated database.

[Tables and Columns in a Database Diagram](#)

Within a database diagram, each table can appear with three distinct features: a title bar, a row selector, and a set of property columns.

Title Bar The title bar shows the name of the table. If another user owns the table, then that user's name appears in parentheses at the end of the table name.



If you have modified a table and have not yet saved it, an asterisk (*) appears at the end of the table name to indicate unsaved changes.

 or 

Row Selector You can click the row selector  to select a database column in the table. The row selector displays a key symbol  if the column is in the table's primary key.

Property Columns The set of property columns is visible only in the certain views of your table. You can view a table in any of four different views to help you manage the size and layout of your diagram.

[Relationships in a Database Diagram](#)

Within a database diagram, each relationship can appear with three distinct features: endpoints, a line style, and related tables.

Endpoints The endpoints of the line indicate whether the relationship is one-to-one or one-to-many. If a relationship has a key at one endpoint and a figure-eight at the other, it is a one-to-many relationship. If a relationship has a key at each endpoint, it is a one-to-one relationship.

Line Style The line itself (not its endpoints) indicates whether the database enforces referential integrity for the relationship when new data is added to the foreign-key table. If the line appears solid, the database enforces referential integrity for the relationship when rows are added or modified in the foreign-key table. If the line appears dotted, the database does not enforce referential integrity for the relationship when rows are added or modified in the foreign-key table.

Related Tables The relationship line indicates that a foreign-key relationship exists between one table and another. For a one-to-many relationship, the foreign-key table is the table near the line's figure-eight symbol. If both endpoints of the line attach to the same table, the relationship is a reflexive relationship.

[Show All](#)

About Database Designer considerations for SQL Server databases (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The following guidelines outline Microsoft SQL Server-specific features that you can use.

[SQL Server versions](#)

The Database Designer is designed to support Microsoft SQL Server 2000, but is compatible with earlier versions as well. If you attempt to use a SQL Server 2000 feature when connected to version 7.0 or 6.5 of SQL Server, the server will report an error.

[Object names](#)

In a Microsoft Access project, names of objects:

- Can be up to 64 characters long.
- Can include any combination of letters, numbers, spaces, and special characters except a period (.), an exclamation point (!), an accent grave (`), double quotation mark ("), and brackets ([]).
- Can't begin with a leading space or a pound sign (#) character. Microsoft SQL Server uses a leading pound sign (#) in an object name to indicate that the object is a temporary object.
- Can't include control characters (ASCII values 0 through 31).

[Case sensitivity](#)

Column and table names in a SQL Server database can be stored in uppercase letters, lowercase letters, or a combination of both. For example, a column name

can appear as "LASTNAME," "LastName," or "lastname."

Depending on how SQL Server was installed, databases can be case-sensitive or case-insensitive. If a database is case-sensitive, you must enter owner, table, and column names using the correct combination of uppercase and lowercase characters. If you are using a case-sensitive database, you must think carefully when you refer to a database object by name, since two objects named "CUSTOMER" and "Customer" can exist in the same database.

If the server was installed with a case-insensitive option, you can enter database object names using any combination of uppercase and lowercase characters.

[Tip](#)

To determine the case sensitivity of a server, execute the system stored procedure `sp_server_info`, and then examine the contents of row 18. If the server has been installed with the case-insensitive setting, the option for `sort_order` will be set to **nocase_iso**.

[Full-text indexes](#)

A table can include full-text indexes. Database Designer and Table Designer provide limited support for manipulating tables with full-text indexes. You cannot use Database Designer or Table Designer to create a full-text index, but if you modify a table with full-text indexes, the Database Designer or Table Designer warns you if your modification affects the full-text index. In most cases, however, when you save your modifications, the Database Designer or Table Designer will be able to reestablish the table's full-text indexes.

For more information on full-text indexing, refer to the SQL Server documentation.

[Changing column properties](#)

[Default values for column properties](#)

The following default values are automatically added for a new column property:

Data Type	Description	Default value
Column Name	The name of a column in a table. Column names must conform to your database rules for identifiers and must be unique in the table.	Blank
Data Type	The data type of the column. System- or user-defined data types are acceptable.	Character (char)
Length	The maximum number of digits (for numeric data types) or characters allowed for values in the column.	Differs for different data types (e.g., 10 for Character, 50 for VARBINARY)
Precision	The maximum total number of decimal digits that can be stored, both to the left and to the right of the decimal point.	0
Scale	The maximum number of decimal digits that can be stored to the right of the decimal point. This value must be less than or equal to the precision. Applies only to DECIMAL and NUMERIC data types.	0
Allow Nulls	Whether or not the column can accept null values.	Yes (selected)
Default Value	The value that will be inserted into the column if the user does not make an entry. Default values are ignored for columns with a timestamp data type. If you do not define a default value and a column allows nulls, NULL will be inserted.	Blank
Identity	Whether or not the column will generate incremental values for new rows based on the Identity Seed and Identity Increment settings.	No (not selected)
Identity Seed	The value assigned to the first row in the table. If the Identity setting is No, Identity Seed is blank. If the Identity setting is Yes, Identity Seed defaults to 1.	Blank or 1.
Identity Increment	The value which is added to the Identity Seed and assigned to the second row in the table. Each subsequent row is increased by this value. If the Identity setting is No, Identity Increment is	Blank or 1.

	blank. If the Identity setting is Yes, Identity Increment defaults to 1.	
Is RowGuid	Whether or not the identity column is used by SQL Server as a ROWGUID column.	No
Formula	The formula for a computed column.	Blank
Collation	The collating sequence that SQL Server applies by default to the column whenever the column values are used to sort rows of a query result (SQL Server 2000 only) .	The database default.
Description	The text description of the selected column stored as a SQL Server extended property (Microsoft SQL Server 2000 only).	Blank

[Changing the data type assigned to a column](#)

A column's data type determines what kind of data can be stored in the column. A list of system-defined data types appears in the Data Type column.

You can choose the appropriate data type for the information you want to store in the column. If user-defined data types exist for your database, they appear at the end of the data type list. The system-defined data type that corresponds to the user-defined data type appears in parentheses at the end of the user-defined data type name. For example: "id (varchar)."

[Special considerations for assigning a user-defined data type to a column](#)

- Selecting a data type automatically sets the length, precision, and scale for the column based on the data type's definition. You cannot change these settings for user-defined data types.
- You can change the Allow Nulls setting only if the user-defined data type allows null values.

For information about creating user-defined data types in Microsoft SQL Server, see the SQL Server documentation.

Note Changing the data type recreates the table in the database when you save the table or diagram.

Caution If you change the data type of a column that is related to columns in other tables, then the data type of the related columns must also be changed to preserve referential integrity. When you save the table or diagram, the **Datat Type Change Required** dialog box enables you to automatically change the data type of the related columns.

For information on Microsoft SQL Server data types, see the SQL Server documentation.

 [Microsoft SQL Server 2000 data types](#)

SQL Server 2000 includes the following data type enhancements:

- **sql_variant**

New data type for storing values of different data types within an individual column.

- **bigint**

New data type for storing large (8-byte) integers.

For more information about data types, see your SQL Server documentation.

 [Changing the column length](#)

When you select a data type, the column length is automatically defined. You can reset the length property for a column with a data type of binary, char, nchar, nvarchar, varbinary, or varchar if you want to increase or decrease the length of acceptable values in that column. For columns with other data types, the length is derived from the data type.

 [Changing the column precision](#)

For most data types, the column precision is automatically defined. You can change the column precision for the decimal and numeric data types if you want to redefine the maximum number of digits these columns use. The precision of a numeric column refers to the maximum number of digits used by the selected data type. The precision of a non-numeric column generally refers to either the

maximum length or the defined length of the column.

The Database Designer prevents you from changing the precision of a column whose data type is not decimal or numeric.

[Changing the column scale](#)

When you select a data type, the column scale by default is set to 0. The scale of a numeric column refers to the maximum number of digits to the right of the decimal point. For columns with approximate floating point numbers, the scale is undefined because the number of digits to the right of the decimal point is not fixed.

You can change the scale for a numeric or decimal column if you want to redefine the number of digits that can appear to the right of the decimal point.

[Changing the null option assigned to a column](#)

For each column in your table, you can specify whether to allow null values or disallow null values. A null value, or NULL, is not the same as zero (0) or blank; NULL means that no entry has been made. Its presence usually implies that the value is either unknown or undefined. For example, a null value in the `price` column of the `titles` table of the `pubs` sample database does not mean that the book has no price; it means that the price is unknown or has not been set.

If null values are not allowed, the user entering data in the table must enter a value in the column or the table row cannot be accepted in the database.

Note You cannot change this property on a primary key column. Also, identity columns cannot have null values. That is, you cannot create or modify a column so that its **Identity** setting is Yes and its **Nulls Allowed** setting is Yes.

Assigning a default value to a column

For each column in your table, you can specify a default value that will be entered in the column if the user leaves it blank. If you do not assign a default value and the user leaves the column blank, then:

- If you set the option to allow null values, NULL will be inserted into the

column.

- If you did not set the option to allow null values, the column will remain blank, but you will not be able to save the row until you supply a value for the column.

For text strings, enclose the value in single quotation marks ('); do not use double quotation marks (") because they are reserved for quoted identifiers. For example, type: 98036 or 'Paris, France'.

If your entry in the **Default Value** column replaces a bound default (which is shown without parentheses), the default will be unbound and the new value will replace it.

[Redefining a global default](#)

A global default is one that is defined for a specific database and is shared by columns of different tables. For example, suppose several of your tables have a quantity column. You can define a global default in your database that inserts a value of 1 in the quantity column whenever the user leaves that column blank in any table.

If a global default is bound to a column, you can specify a different default value for that column in a specific table. In such a case, the existing global default is unbound from the column before the new default value is bound to the column.

[Changing a column's identity properties](#)

You can change the identity properties of a column if you want to redefine the sequential numbers that are automatically generated and stored in that column when new rows are added to the table. You can set the identity properties on only one column per table.

Columns that have the identity property contain system-generated sequential values that uniquely identify each row within a table (for example, employee identification numbers). When inserting values into a table with an identity column, Microsoft SQL Server automatically generates the next identifier based on the last used identity value (the identity seed property) and the increment value (the identity increment property) specified during the creation of the column.

The identity property can be set only for a column whose data type is decimal, int, numeric, smallint, bigint, or tinyint and that disallows null values.

For example, suppose you want to automatically generate a 5-digit Order ID for each row added to the orders table, beginning with 10000 and incremented by a value of 10. To do this, you would select the **Identity** property box, type an **Identity Seed** of 10000, and type an **Identity Increment** of 10.

If you change any of the identity properties for a table, the existing identity values will be preserved. Your new settings apply only to new rows that are added to the table.

Note If an identity column exists for a table with frequent deletions, gaps can occur between identity values. If you want to avoid such gaps, do not use the identity property.

Discard changes made in Database Designer or Table Designer (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can discard changes in your database diagram that you do not want to save in the database.

1. In the Database Designer or Table Designer, click **Save** .
2. A message prompts you to save your changes. Choose **No**.

Caution If you have other open database diagrams or table design windows connected to the same database, the modifications that you discard in this manner remain in your local memory. To remove these pending modifications from memory, you must close all database diagrams and table design windows connected to the same database.

[Show All](#)

About large database projects (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Overview of large database projects](#)

When you use Microsoft Access as part of a large software development effort, you can encounter several noteworthy situations:

- **Multiple People Designing a Single Database** Several users can connect to a database and use Access to alter the database design.
- **Evolutionary Development of a Deployed Database** After a database is deployed, design changes can become necessary as users expand the set of tasks they want to perform with the data.
- **Multiple Versions of a Particular Database** Large projects can include many databases. Even a project with a single database can have several copies of it— a development database, a test database, and a deployed production database. During the lifetime of a deployed database application, changes and improvements are first made in the development database, then propagated to the test database, then propagated to the production database.

 [Multiuser environments](#)

 [Overview of multiuser environments](#)

A multiuser environment is one in which other users can connect and make changes to the same database that you are working with. As a result, several users might be working with the same database objects at the same time. Thus, a multiuser environment introduces the possibility of your database diagrams being affected by changes made by other users, and vice versa. Such changes could include changes to copies of your database diagrams, other users' database diagrams that share database objects with your database diagrams, or the underlying database.

A key issue when working with databases in a multiuser environment is access permissions. The permissions you have for the database determine the extent of the work you can do with the database. For example, to make changes to objects in a database, you must have the appropriate write permissions for the database. For more information about permissions in your database, see the SQL Server documentation.

As one of multiple users, you may need to address any of the following issues when working with the Database Designer:

- Ownership of database objects
- Diagrams affected by another user's changes
- Database objects deleted by another user

[Ownership of database objects](#)

Each database object is owned by either a user or a role. If the owner is a role, the object is co-owned by every user belonging to that role.

Each object also has privileges associated with it. A privilege grants a particular authority to an object for either an individual user or for all users belonging to a particular role.

Depending on the privileges defined in the database, you can view objects that you do not own. For example, you can include on a database diagram an object owned by another user. When such an object appears on a database diagram or in the Table Designer, it is labeled with the owner name as well as the object name.

[Diagrams affected by another user's changes](#)

In a multiuser environment, your database diagram can be affected by changes other users have saved to:

- Your database diagram, which other users changed since you opened the diagram.
- Database objects shared between database diagrams.
- The database.

For example, your database diagram might contain a table that another user deleted or renamed. In such a case, your database diagram will no longer reflect the current state of the database. When you attempt to save your database diagram or selected tables, the **Database Changes Detected** dialog box notifies you that the database has been updated since you opened your diagram.

This dialog box also displays a list of database objects that will be affected as a result of saving your database diagram or selection. At this point, you can take one of these actions:

- **Save your database diagram or selection and update the database with all the changes in the list.** This action will also affect other database diagrams that share the same database objects.
For example, suppose you edit the `au_id` column in the `titleauthors` table on your database diagram and another user's database diagram contains the `authors` table which is related to the `titleauthors` table by the `au_id` column. Saving your database diagram will affect the other user's database diagram. Similarly, suppose that another user defined a check constraint for the `qty` column in the `sales` table. If you delete the `qty` column and save the `sales` table, the other user's check constraint will be affected.
- **Cancel the save action.** You can then close the database diagram without saving it. When you reopen the database diagram, it will be in synch with the database.
- **Save a list of the changes.** You can save the list of database changes shown in the **Database Changes Detected** dialog box to a text file so that you can investigate the cause of other users' changes.
For example, if another user edited a table that you marked for deletion, you may want to research whether the table should be deleted before updating the database.

[Database objects deleted by another user](#)

In a multiuser environment, deleting database objects from the database can affect other users. If another user deletes a database object that appears in your database diagram, the effect of the deletion on your database diagram depends on where the object was deleted.

**Where the
object was**

How your database diagram is affected

deleted

In another copy of your database diagram	The deleted object will be removed from your database diagram next time you open it. If your database diagram is open when the deletion occurs, it still shows the deleted item.
In a different database diagram	The deleted object will be removed from your database diagram next time you open it. If your database diagram is open when the deletion occurs, it still shows the deleted item. If your database diagram has no unsaved changes to that object, the object will be removed from your database diagram the next time you open it.
In the database	If your database diagram contains unsaved changes to that object, the object will remain in your database diagram. You can recreate it in the database by saving your database diagram.

Note Recreating a deleted object creates a new definition of that object in the database; it does not restore the data that was deleted when the object was deleted.

For example, if the `titles` table is deleted in the database and you have unsaved changes to the `titles` table in your database diagram, then saving your database diagram will create a new `titles` table in the database. Any data that existed in the `titles` table before it was deleted is not restored.

[Issues of database evolution](#)

If you change the structure of a deployed database, you must take special care that your alteration is compatible with the existing data and database structure. You might need to take special steps when you make the following modifications:

- **Adding a Constraint** If you add a constraint, the database might already contain data that does not satisfy it. When you try to save the new constraint, the **Save Incomplete** dialog box informs you that the database could not create the constraint. To force the database to accept the new constraint, you can clear the **Check existing data on creation** check box.
- **Adding a Relationship** If you add a relationship, the database might

already contain rows of the foreign-key table that do not have corresponding rows in the primary-key table. That is, the existing data might not satisfy referential integrity. When you try to save the new relationship, the **Save Incomplete** dialog box informs you that the database could not save the revised foreign-key table. To force the database to accept the modification, you can clear the **Check existing data on creation** check box.

- **Modifying a Table Contributing to an Indexed View** If you modify a table that contributes to a Microsoft SQL Server indexed view, the indexes on the view will be lost. See the SQL Server documentation for information on recreating indexes.

No matter how you alter the database design, you should retain a history of the alterations.

[Design development, test, and production databases](#)

If you have two databases with identical structure, you can make changes in one database and propagate those changes to the other. For example, if you have a personal development database and a group-wide test database, you can modify the development database, then propagate those changes to the test database.

[Show All](#)

Add labels to relationship lines in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Add labels to relationship lines in a diagram](#)

1. Right-click the database diagram.
2. From the shortcut menu, choose **Show Relationship Labels**.

[Rename relationship labels](#)

1. Select the relationship line you want to rename.
2. Right-click the relationship line and select **Properties** from the shortcut menu.
3. Select the **Relationships** tab.
4. In the **Relationship name** box, type a new name for the relationship label.

Note You cannot change the font for relationship labels.

If relationship line labels are not displayed, you can still see the relationship line label for a given relationship.

[View a label for a specific relationship line](#)

1. In the database diagram, move the mouse over the relationship line.
2. The Tooltip will show the relationship label.

Adjust the table size in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can adjust the size of a table in the database diagram in any of its five views. For example, you can enlarge a table in Column Properties view to see all of the properties for its columns without having to scroll the table horizontally.

A minimized table can be resized only horizontally.

1. Select the table(s) you want to resize.
2. Right-click the table and choose **Autosize Selected Tables** from the shortcut menu.

The table and the columns within the table are adjusted to show all the information in the displayed columns.

[Show All](#)

Annotate database diagrams (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can put notes into your database diagram, to describe the database diagram and its objects. These can include headings, descriptions of the tables and their contents, and any other information you want to place in the database diagram.

[Add annotations to a database diagram](#)

1. Right-click the database diagram.
2. From the shortcut menu, choose **New Label**. A blank text box appears in the diagram at the location where you right-clicked.
3. Type the annotation in the text box.

[Change the font of the annotation](#)

1. Select the text annotation box whose font you want to change.
2. Right-click inside the text annotation box and choose **Set Label Font** from the shortcut menu.

Note All text in the text annotation box will be changed to the new font.

[Arrange the text annotation in a database diagram](#)

1. Select the text annotation box.
2. Drag the box into position in the diagram.

Note You can only use the [ANSI character set](#) in an annotation; you cannot use [Unicode](#) characters.

[Tip](#)

You can use annotations to describe the relationships between tables in your diagram. However, you can also generate relationship labels automatically.

[Show All](#)

Arrange tables automatically in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can automatically rearrange your database diagram to display the table or tables you want to work on, or you can rearrange the entire diagram in a way that best displays all the tables and their relationships. This is especially useful if you've moved the tables in the diagram, or added new tables, and want to clean up the layout of the diagram.



[Arrange all tables in a database diagram automatically](#)

1. Right-click the database diagram.
2. From the shortcut menu, select **Arrange Tables**.



[Arrange selected tables in a database diagram automatically](#)

1. Select the tables you want to arrange.
2. Right-click a table and choose **Arrange Selection** from the shortcut menu.

Create a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can use a database diagram to create, edit, or delete database tables. You can also use a database diagram to see how your tables relate to each other. When you create a new database diagram, you are presented with a blank diagram surface to work on.

1. In the Database window, click **Database Diagrams**  under **Objects**, and choose the **New** button on the Database window toolbar.

A blank database diagram appears with the **Add Table** dialog box.

2. To add a table to the database diagram, select a table, and then click **Add**.

-or-

To create a new table in the diagram, right-click anywhere inside the database diagram and then choosing **New Table** from the shortcut menu.

3. Arrange the objects in your database diagram.

If you want to move objects to different parts of your diagram— for example, to align the tables or reduce the number of overlapping relationship lines— first select the objects, and then drag the selection to its new location.

- To select a table, click its title bar.
- To select a set of tables, hold down the CTRL key as you click inside the title bar of each table in the set.
- To select a set of tables grouped together, position the pointer near one of the tables and then drag a rectangle around all the tables in the set.

If you select multiple tables, the relationship lines that connect them are automatically included in the selection.

If you select a relationship line, you can slide a line segment to a location that is parallel to its original position. If the relationship line has more than one segment, you can drag a corner diagonally to change both its vertical and horizontal position.

Interactions among database diagrams and table design windows (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you connect to a database and begin designing or modifying a database diagram or a table, Microsoft Access retains your work in memory. That is, Access does not transmit your work to the database until you explicitly save the work there. Regardless of how many database diagrams you open or how many tables you design, Access retains a single in-memory model of the database structure. There are several ramifications:

- **You can experiment with different object definitions**

Because your modifications are not saved to the database immediately, you can experiment to see how a proposed modification will affect the database. When you complete your modifications, you can either save your changes to the database or discard your changes.

- **Your modifications can appear in many database diagrams or table design windows**

When you modify a database object, every open database diagram containing that object will reflect that modification. For example, if you add a column to a table, the new column appears on every open database diagram containing that table. If you modify an object and later add that object to another database diagram, the added object reflects the modifications—even if you have not yet saved the modifications to the database.

- **Your modifications can exist in memory only**

The in-memory model of the database structure endures until you close all

database diagrams and table design windows for that database. Thus, it is possible that the in-memory model retains modifications that are not visible on any open database diagrams. Even if you remove a modified object from the only open database diagram, the modification remains in memory. If you later add the same object to any database diagram, the modification will be visible.

[Show All](#)

Interactions between database diagrams, table design windows, and the database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).



[Effects before saving design modifications to the database](#)

When you connect to a database and begin designing or modifying a database diagram or a table, Microsoft Access retains your work in memory. That is, Access does not transmit your work to the database until you explicitly save the work there. Regardless of how many database diagrams you open or how many tables you design, Access retains a single in-memory model of the database structure. There are several ramifications:

- **You can experiment with different object definitions**

Because your modifications are not saved to the database immediately, you can experiment to see how a proposed modification will affect the database. When you complete your modifications, you can either save your changes to the database or discard your changes.

- **Your modifications can appear in many database diagrams or table design windows**

When you modify a database object, every open database diagram containing that object will reflect that modification. For example, if you add a column to a table, the new column appears on every open database diagram containing that table. If you modify an object and later add that object to another database diagram, the added object reflects the modifications—even if you have not yet saved the modifications to the database.

- **Your modifications can exist in memory only**

The in-memory model of the database structure endures until you close all database diagrams and table design windows for that database. Thus, it is possible that the in-memory model retains modifications that are not visible on any open database diagrams. Even if you remove a modified object from the only open database diagram, the modification remains in memory. If you later add the same object to any database diagram, the modification will be visible.

[Effects after saving design modifications to the database](#)

When you save your work from a database diagram or Table Designer, the Save operation transmits your additions and modifications to the database as a series of SQL statements. Several issues can arise during this process:

- **Your modifications can require the deletion and re-creation of some database objects.**

Many databases support a limited set of ways you can modify an existing object. A typical limitation disallows the reordering of columns. But you can use the Table Designer to reorder a table's columns, even if you are connected to a database that does not directly accommodate the reordering of columns, such as Microsoft SQL Server. To circumvent the database limitation, the Save operation actually creates a new table with the desired characteristics and deletes the existing table.

The limitation disallowing column reordering is merely an example; there are several such limitations. The Save operation uses the delete-and-re-create technique to circumvent many of these limitations. When using the delete-and-re-create strategy, the Save operation preserves any rows of the table, as well as foreign-key relationships and other settings.

- **Your modifications might indirectly affect more database objects than you realize.**

In some cases, the Save operation will delete and re-create several tables because those tables are related to a table you have modified.

For example, if you change the data type of the `au_id` column in the `titleauthor` table, the Save operation must preserve the existing

relationships from titleauthor to titles and authors. To preserve these relationships, the Save operation must re-create all three tables and both relationships.

Because the preservation of relationships can induce delete-and-re-create operations on several related tables, the save operation can yield the **Save** Dialog Box, which warns you about which tables will be affected by the save operation. The **Save** dialog box appears before your modifications are transmitted to the database, so you can see which table will be affected before choosing to proceed with or cancel the save operation.

- **Another user might have recently modified some database objects that you are now trying to modify.**

As you use a database diagram or the Table Designer to modify database objects, another user might be modifying the same objects. If that other user saves his or her work before you do, your subsequent Save operation will detect that recent changes to the database have affected the objects you are trying to modify. To help you decide how to proceed, the **Database Changes Detected** Dialog Box appears. The **Database Changes Detected** dialog box appears before your changes are transmitted to the database, so you can see which recent modifications you will be overwriting if you choose to proceed with the save operation.

- **Before transmitting your modifications, Microsoft Access tries to anticipate problems with the Save operation.**

There are several reasons why your save operation might fail. For example, if the save operation induces an implicit delete-and-re-create operation on a table that you do not have permission to delete, your operation can fail. Similarly, if one of your modifications would yield an index that exceeds the maximum size for indexes, your operation can fail. Although the database itself is the final arbiter of which modifications are legal and which are not, Microsoft Access will try to anticipate how your modifications might fail. To warn you of these potential problems, the **Validation Warnings** Dialog Box appears. It appears before your modifications are transmitted, so you can choose whether or not to proceed.

- **The implicit delete-and-re-create operation does not repopulate full-**

text indexes.

If your save operation induces an implicit delete-and-re-create operation on a table that has full-text indexing, the index will be re-created, but the full-text catalog will not be repopulated. You should repopulate the full-text catalog after all of your modifications are complete. To remind you of any full-text indexes that need to be repopulated, the **Post-Save Notifications** Dialog Box appears. It appears after the database actually attempts to satisfy your transmitted save operation.

- **The database might complete only a portion of the transmitted Save operation.**

Remember, the database is the final arbiter of which modifications are valid and which modifications you have permission to complete. When you transmit your save operation to the database, some objects might be modified and some might not, based on permissions, constraints, or any of several other factors. To inform you of any failures in the save operation, the **Post-Save Notifications** Dialog Box appears.

Merge subdiagrams into a master database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Merge a set of subdiagrams into a master diagram to consolidate multiple diagrams for a large group of tables that are logically related. For example, as you add new modules to an existing application, you can create one diagram for each new module. You can then merge all the diagrams into the master diagram.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the master database diagram you want to open, and then click **Design** on the Database window toolbar.
2. Open the subdiagram that you want to merge into the master database diagram.
3. Select **Tile Horizontally** or **Tile Vertically** from the **Window** menu so that both diagrams are visible.
4. In the subdiagram, choose **Select All** from the **Edit** menu.
5. Position the pointer on the title bar of one of the selected tables and drag the selection to the master diagram. If your selection contains a table that is already displayed on the master diagram, it will not be duplicated. A diagram cannot contain duplicate copies of a table.
6. Repeat steps 2 through 5 for each subdiagram that you want to merge into the master diagram.

When you consolidate a group of tables into one database diagram by merging subdiagrams into a master database diagram, the layout of the individual subdiagrams is maintained within the master database diagram.

[Show All](#)

Reconcile a database diagram with a modified database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

You save your database diagram when you are ready to update the database to match your database diagram. However, if other users have updated the database since you opened your database diagram, their changes might affect your database diagram and vice versa. In such cases, saving your database diagram will automatically:

- Recreate database objects that are referenced in your database diagram but that another user deleted.
- Preserve triggers that were added to a table; triggers that were deleted from a table are not preserved.
- Delete an object you deleted in your database diagram, even if another user edited that object.

Saving your database diagram will reconcile the database with your database diagram by overwriting other users' changes so that the database will match your database diagram.

[Reconcile your database diagram with a modified database](#)

1. In the Database Designer, click **Save** .

If you have not previously saved your database diagram, type a name for the database diagram in the **Save As** dialog box and choose **OK**.

2. The **Save** dialog box lists the tables that will be affected when you save your database diagram. Choose **Yes** to continue.
3. The **Save** dialog box lists the objects that were modified and will be

changed to match your database diagram. Choose **Yes** to save the database diagram and accept the list of changes.

Note If your database diagram contains tables and columns that were deleted in the database, only their definitions are recreated in the database when you save your database diagram. This process does not restore any data that existed in these objects before their deletion.

Save an entire database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

By saving a database diagram, you can save all the changes you made to it, including any changes you made to the tables, columns and other database objects.

1. In the Database Designer, click **Save** .
2. If this is a new diagram that you have never before saved, the **Save As** dialog box appears. Type a name for the database diagram.

If you made changes to tables in an existing database diagram, the **Save** dialog box appears and displays a list of changes that will be saved in the database when you save your database diagram.

3. Click **Yes** to update the database to match your database diagram.

Save selected tables on a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can save a specific table or a set of tables if you do not want to save all the changes you made in a database diagram.

1. In your database diagram, select the tables you want to save.
2. In the Database Designer, click **Save** .
3. The **Save** dialog box displays the list of tables that will be updated in the database when you save your selection.

Choose **Save Text File** if you want to save the list of tables in a text file in the project directory before continuing. When a message box displays the name of the saved text file, choose **OK**.

4. In the **Save** dialog box, confirm the list of tables and choose **Yes** to save these tables.

Note The list of tables may contain tables in addition to those selected. For example, if you change the data type of a column that participates in a relationship with another table, both tables will be included in this list.

[Show All](#)

Select multiple objects in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can perform actions on a group of objects by selecting multiple objects in a database diagram. You can select multiple objects that are adjacent to each other or that are spatially separated from each other by other objects.

[Select all objects in the diagram](#)

- Right-click the diagram and choose **Select All** from the shortcut menu.

[Select a group of adjacent tables](#)

- Hold down the SHIFT key while you click each table that you want to add to your selection.

–or–

- Point to the diagram background and drag a rectangle around the tables you want to select. Each table does not have to be fully contained in the rectangle to be selected.

[Select a group of spatially separated tables](#)

- Hold down the CTRL key while you click each table that you want to add to your selection. If you click the title bar of an already selected table, that table is removed from the selection.

[Select a group of adjacent columns](#)

- Hold down the SHIFT key while you click the row selector  of the first and last columns that you want to select. This method selects every column

from the column that displays the selected row indicator  to the last row selector you clicked.

[Select a group of spatially separated columns](#)

- Hold down the CTRL key while you click the row selector  of each column that you want to add to your selection. If you click the row selector of an already selected column , that column is removed from the selection.

[Select a relationship line](#)

- Click the relationship line.

You can select only one relationship line at a time. For example, you can select a relationship line when you want to change its properties. When you select a group of tables, the relationship lines between the tables are automatically included in the selection.

[Show All](#)

Subdivide a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Database diagrams can range in complexity from just one table to hundreds of tables. When you're working with a large number of tables, you can create a "master" database diagram that contains all the tables you need, and then divide the master diagram into many smaller database diagrams or "subdiagrams."

Subdivide your database diagram when you want to isolate a group of the tables. For example, if you are designing an application with many modules, you can create one master database diagram with all the tables that your application needs. You can then subdivide the master database diagram so that each subdiagram contains only the tables pertaining to a specific module.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the master database diagram you want to open, and then click **Design** on the Database window toolbar.
2. Create a new database diagram.

 [How?](#)

You can use a database diagram to create, edit, or delete database tables. You can also use a database diagram to see how your tables relate to each other. When you create a new database diagram, you are presented with a blank diagram surface to work on.

1. In the Database window, click **Database Diagrams**  under **Objects**, and choose the **New** button on the Database window toolbar.

A blank database diagram appears with the **Add Table** dialog box.

2. To add a table to the database diagram, select a table, and then click **Add**.

-or-

To create a new table in the diagram, right-click anywhere inside the database diagram and then choosing **New Table** from the shortcut menu.

3. Arrange the objects in your database diagram.

If you want to move objects to different parts of your diagram— for example, to align the tables or reduce the number of overlapping relationship lines— first select the objects, and then drag the selection to its new location.

- To select a table, click its title bar.
- To select a set of tables, hold down the CTRL key as you click inside the title bar of each table in the set.
- To select a set of tables grouped together, position the pointer near one of the tables and then drag a rectangle around all the tables in the set.

If you select multiple tables, the relationship lines that connect them are automatically included in the selection.

If you select a relationship line, you can slide a line segment to a location that is parallel to its original position. If the relationship line has more than one segment, you can drag a corner diagonally to change both its vertical and horizontal position.

3. In the master database diagram, select the group of tables you want to isolate in the subdiagram.

[How?](#)

You can perform actions on a group of objects by selecting multiple objects in a database diagram. You can select multiple objects that are adjacent to each other or that are spatially separated from each other by other objects.

[Select all objects in the diagram](#)

- Right-click the diagram and choose **Select All** from the shortcut menu.

[Select a group of adjacent tables](#)

- Hold down the SHIFT key while you click each table that you want to add to your selection.

–or–

- Point to the diagram background and drag a rectangle around the tables you want to select. Each table does not have to be fully contained in the rectangle to be selected.

[Select a group of spatially separated tables](#)

- Hold down the CTRL key while you click each table that you want to add to your selection. If you click the title bar of an already selected table, that table is removed from the selection.

[Select a group of adjacent columns](#)

- Hold down the SHIFT key while you click the row selector of the first and last columns that you want to select. This method selects every column from the column that displays the selected row indicator to the last row selector you clicked.

[Select a group of spatially separated columns](#)

- Hold down the CTRL key while you click the row selector of each column that you want to add to your selection. If you click the row selector of an already selected column , that column is removed from the selection.

[Select a relationship line](#)

- Click the relationship line.

You can select only one relationship line at a time. For example, you can select a relationship line when you want to change its properties. When you select a group of tables, the relationship lines between the

tables are automatically included in the selection.

4. Copy or drag the tables into the new database diagram.

Because each table that appears in a database diagram is merely a reference to a database table, you need to edit the table in only one database diagram in order for your changes to be immediately reflected in each database diagram in which the table appears.

[Show All](#)

Zoom a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Change the zoom level of your database diagram to control how much of the diagram you can see on your screen at once.

[Zoom a database diagram](#)

- Select a setting from the **Zoom** submenu on the **View** menu.

Note To edit tables, you must set the zoom level to 50% or higher. To create relationships, you must set the zoom level to 25% or higher.

[Zoom to a selected object](#)

- Select an object in the diagram and choose **Zoom Selection** from the **Zoom** submenu on the **View** menu.

[Tip](#)

To center the entire diagram in the window, use **Select All** to highlight all objects in the database diagram and then choose **Zoom Selection** from the **Zoom** submenu on the **View** menu. This will zoom your diagram to fit in the current diagram window size.

[Show All](#)

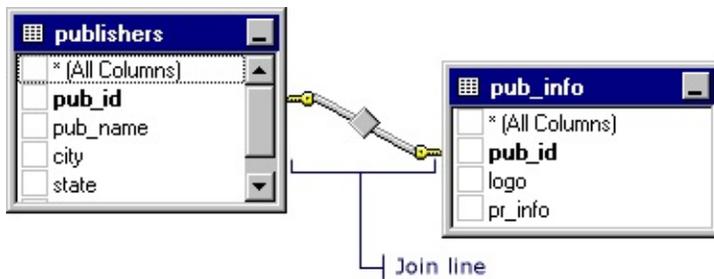
How the Query Designer represents joins (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

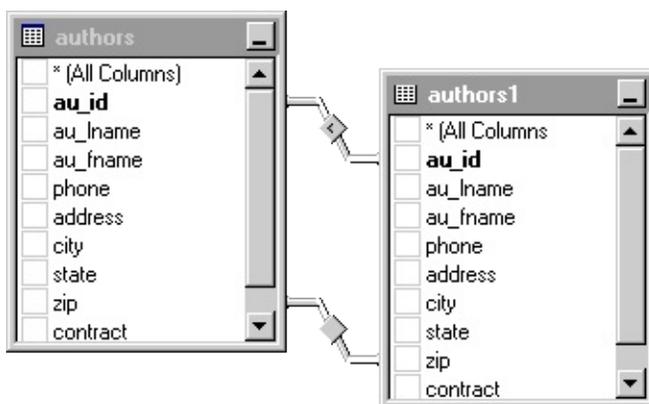
If tables are joined, the Query Designer represents the join graphically in the Diagram Pane and by using SQL syntax in the SQL Pane.

[Diagram Pane](#)

In the **Diagram** pane the Query Designer displays a join line between the data columns involved in the join. The Query Designer displays one join line for each join condition. For example, the following illustration shows a join line between two tables that are joined:



If tables are joined using more than one join condition, the Query Designer displays multiple join lines, as in the following example:

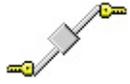


If the joined data columns are not displayed (for example, the rectangle representing the table, view, function, or subquery is minimized or the join involves an expression), the Query Designer places the join line at the title bar of the rectangle representing the table, view, function, or subquery.

The shape of the icon in the middle of the join line indicates how the tables, views, or functions are joined. If the join clause uses an operator other than equal (=), the operator appears in the join line icon. The following table lists the icons that appear in the join line.

Join line icon	Description
	Inner join (created using an equal sign).
	Inner join based on the "greater than" operator.
	Outer join in which all rows from the table represented on the left will be included, even if they do not have matches in the related table.
	Outer join in which all rows from the table represented on the right will be included, even if they do not have matches in the related table.
	Full outer join in which all rows from both tables will be included, even if they do not have matches in the related table.

The symbols on the ends of the join line indicate the type of join. The following table lists the types of joins and the icons displayed on the ends of the join line.

Icon on ends of join line	Type of join
	One-to-one join.
	One-to-many join.
	Query Designer cannot determine the join type. This situation occurs most often when you have created a join manually.

 [SQL Pane](#)

A join can be expressed in a number of ways in an SQL statement. The exact syntax depends on the database you are using and on how you have defined the

join.

Syntax options for joining tables include:

- **JOIN qualifier for the FROM clause.** The keywords INNER and OUTER specify the join type. This syntax is standard for ANSI 92 SQL.

For example, if you join the publishers and pub_info tables based on the pub_id column in each table, the resulting SQL statement might look like this:

```
SELECT *
FROM publishers INNER JOIN pub_info ON
    publishers.pub_id = pub_info.pub_id
```

If you create an outer join, the words LEFT OUTER or RIGHT OUTER appear in place of the word INNER.

- **WHERE clause compares columns in both tables.** A WHERE clause appears if the database does not support the JOIN syntax (or if you entered it yourself). If the join is created in the WHERE clause, both table names appear in the FROM clause.

For example, the following statement joins the publishers and pub_info tables.

```
SELECT *
FROM publishers, pub_info
WHERE publishers.pub_id = pub_info.pub_id
```

Note Microsoft SQL Server databases support *= and =* syntax.

About joining tables automatically (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you add two or more tables to a query, the Query Designer attempts to determine if they are related. If they are, the Query Designer automatically puts join lines between the rectangles representing the tables, views, or functions.

The Query Designer will recognize tables as joined if:

- The database contains information that specifies that the tables are related.
- If two columns, one in each table, have the same name and data type. The column must be a primary key in at least one of the tables. For example, if you add `employee` and `jobs` tables, if the `job_id` column is the primary key in the `jobs` table, and if each table has a column called `job_id` with the same data type, the Query Designer will automatically join the tables.

Note The Query Designer will create only one join based on columns with the same name and data type. If more than one join is possible, the Query Designer stops after creating a join based on the first set of matching columns that it finds.

- The Query Designer detects that a search condition (a WHERE clause) is actually a join condition. For example, you might add the tables `employee` and `jobs`, then create a search condition that searches for the same value in the `job_id` column of both tables. When you do, the Query Designer detects that the search condition results in a join, and then creates a join condition based on the search condition.

If the Query Designer has created a join that is not suitable to your query, you can modify the join or remove it. If the Query Designer does not automatically join the tables in your query, you can create a join yourself.

Check existing data when creating a relationship (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Select the option to check existing data when you create a relationship if the foreign key constraint should apply to existing data as well as to new data.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the table that the foreign key constraint is attached to.
3. Right-click the table and select **Relationships**.
4. Select the relationship from the **Selected relationship** list.
5. Select the **Check existing data on creation** check box.

The foreign key constraint is applied when you save the table or diagram. If any constraint violations are encountered during the save process, the table cannot be saved.

Copy primary key column properties to a foreign key column (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you copy a column from one table to another table, the column name and many of the other properties you defined for that column in the original table are copied to the new location. You can copy column properties from a primary key column to a foreign key column if you want to relate the two columns. Related columns must have the same data type (or data types that can be related) and length property settings.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the primary key columns that you want to copy. Primary key columns are indicated by a key symbol  in their row selector.
3. Copy the columns to the foreign key table using the **Copy** and **Paste** commands on the **View** menu.
4. Drag a relationship line from the columns in the primary key table to the same columns in the foreign key table. Microsoft Access shows the **Create Relationship** dialog box. For more information on each option, click **Help**.

[Show All](#)

Create an outer join in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

By default, the Query Designer creates an inner join between tables. If you want to include data rows in the result set that do not have a match in the joined table, you can create an outer join.

When you create an outer join, the order in which tables appear in the SQL statement (as reflected in the SQL pane) is significant. The first table you add becomes the "left" table and the second table becomes the "right" table. (The actual order in which the tables appear in the Diagram pane is not significant.) When you specify a left or right outer join, you are referring to the order in which the tables were added to the query and to the order in which they appear in the SQL statement in the SQL pane.

 [Create an outer join](#)

1. Create the join.

 [How?](#)

When you add two (or more) tables to a query, the Query Designer attempts to join them based on common data or on information stored in the database about how tables are related. However, if the Query Designer has not joined the tables automatically, or if you want to create additional join conditions between tables, you can join tables manually.

You can create joins based on comparisons between any two columns, not just columns that contain the same information. For example, if your database contains two tables, `titles` and `roysched`, you can compare values in the `ytd_sales` column of the `titles` table against the `lorange` and `hirange` columns in the `roysched` table. Creating this join would allow you to find titles for which the year-to-date sales falls between the low and high ranges for the royalty payments.

Tip

Joins work fastest if the columns in the join condition have been indexed. In some cases, joining on unindexed columns can result in a slow query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add to the Diagram pane the objects you want to join.

How?

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for a table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL** to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Drag the name of the join column in the first table, view, or function and drop it onto the related column in the second table, view, or function. You cannot base a join on text, ntext, or image columns.

Note The join columns must be of the same (or compatible) data types. For example, if the join column in the first table is a date, you must relate it to a date column in the second table. On the other hand, if the first join column is an integer, the related join column must also be of an integer data type, but it can be a different size. The Query Designer will not check the data types of the columns you use to create a join, but when you execute the query, the database will display an error if the data types are not compatible.

4. If necessary, change the join operator; by default, the operator is an equal sign (=).

[How?](#)

By default, the Query Designer joins tables using an equal sign (an equijoin), which matches values in the two join columns. If you want, you can change the operator used to compare values in the join columns.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the join line you want to modify, and then choose **Properties** from the shortcut menu.
3. In the **Join Line** tab of the **Properties** dialog box, select a new operator from the list.

Note You can also change the operator in the SQL statement in the SQL pane.

The Query Designer adds an INNER JOIN clause to the SQL statement in the SQL pane.

Select the join line in the **Diagram** pane, and then from either the **Query** menu or the shortcut menu, choose **Select All Rows from *table***, selecting the command that includes the table whose extra rows you want to include.

–or–

In the **Diagram** pane, right-click the join line you want to change to an outer join, and then choose **Properties** from the shortcut menu. Under **Include rows** in the Join Line tab of the **Properties** dialog box, choose the option specifying the table from which you want to include all rows.

- Choose the first table to create a left outer join.
- Choose the second table to create a right outer join.
- Choose both tables to create a full outer join.

When you specify an outer join, the Query Designer modifies the join line to indicate an outer join.

In addition, the Query Designer modifies the SQL statement in the SQL pane to reflect the change in join type, as shown in the following statement:

```
SELECT employee.job_id, employee.emp_id,  
       employee.fname, employee.minit, jobs.job_desc  
FROM employee LEFT OUTER JOIN jobs ON  
       employee.job_id = jobs.job_id
```

 [Example of an outer join query used to find rows that violate foreign key constraints](#)

Because an outer join includes unmatched rows, you can use it to find rows that violate foreign key constraints. To do so, you create an outer join and then add a search condition to find rows in which the primary key column of the rightmost table is null. For example, the following outer join finds rows in the employee table that do not have corresponding rows in the jobs table:

```
SELECT employee.emp_id, employee.job_id  
FROM employee LEFT OUTER JOIN jobs  
       ON employee.job_id = jobs.job_id  
WHERE (jobs.job_id IS NULL)
```

[Show All](#)

Create a self-join in a query automatically (ADP)

If a table has a reflexive relationship in the database, you can join it to itself automatically.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add to the Diagram pane the table, view, or function you want to work with.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL**  to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Add the same table, view, or function again, so that the **Diagram** pane shows the same table, view, or function twice within the Diagram pane. The Query Designer assigns an alias to the second instance by adding a sequential number to the table, view, or function name. In addition, the Query Designer creates a join line between the two rectangles representing the two different ways the table, view, or function participates in the query.

[Show All](#)

Create a self-join in a query manually (ADP)

[Background information](#)

You can join a table to itself even if the table does not have a reflexive relationship in the database. For example, you can use a self-join to find pairs of authors living in the same city.

As with any join, a self-join requires at least two tables. The difference is that, instead of adding a second table to the query, you add a second instance of the same table. That way, you can compare a column in the first instance of the table to the same column in the second instance, which allows you to compare the values in a column to each other. The Query Designer assigns an alias to the second instance of the table.

For example, if you are creating a self-join to find all pairs of authors within Berkeley, you compare the `city` column in the first instance of the table against the `city` column in the second instance. The resulting query might look like the following:

```
SELECT
  authors.au_fname,
  authors.au_lname,
  authors1.au_fname AS Expr2,
  authors1.au_lname AS Expr3
FROM
  authors
INNER JOIN
  authors authors1
ON authors.city
= authors1.city
WHERE
  authors.city = 'Berkeley'
```

Creating a self-join often requires multiple join conditions. To understand why, consider the result of the preceding query:

```
Cheryl Carson Cheryl Carson
Abraham Bennet Abraham Bennet
Cheryl Carson Abraham Bennet
Abraham Bennet Cheryl Carson
```

The first row is useless; it indicates that Cheryl Carson lives in the same city as Cheryl Carson. The second row is equally useless. To eliminate this useless data, you add another condition retaining only those result rows in which the two author names describe different authors. The resulting query might look like this:

```
SELECT
  authors.au_fname,
  authors.au_lname,
  authors1.au_fname AS Expr2,
  authors1.au_lname AS Expr3
FROM
  authors
INNER JOIN
  authors authors1
ON authors.city
= authors1.city
AND authors.au_id
<> authors1.au_id
WHERE
  authors.city = 'Berkeley'
```

The result set is improved:

```
Cheryl Carson Abraham Bennet
Abraham Bennet Cheryl Carson
```

But the two result rows are redundant. The first says Carson lives in the same city as Bennet, and the second says the Bennet lives in the same city as Carson. To eliminate this redundancy, you can alter the second join condition from “not equals” to “less than”. The resulting query might look like this:

```
SELECT
  authors.au_fname,
  authors.au_lname,
  authors1.au_fname AS Expr2,
  authors1.au_lname AS Expr3
FROM
  authors
```

```
INNER JOIN
authors authors1
ON authors.city
= authors1.city
AND authors.au_id
< authors1.au_id
WHERE
authors.city = 'Berkeley'
```

And the result set looks like this:

Cheryl Carson Abraham Bennet

[Create a self-join manually](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add to the Diagram pane the table, view, or function you want to work with.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane

accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL** to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Add the same table again, so that the Diagram pane shows the same table, view, or function twice within the Diagram pane.

The Query Designer assigns an alias to the second instance by adding a sequential number to the table name. In addition, the Query Designer creates a join line between the two occurrences of the table, view, or function within the Diagram pane.

4. Right-click the join line, choose **Properties** from the shortcut menu, and then change the comparison operator between the primary keys as required. For example, you might change the operator to less than (<).
5. Create the additional join condition (for example, authors.zip = authors1.zip) by dragging the name of the primary join column in the first occurrence of the table, view, or function and dropping it on the corresponding column in the second occurrence.
6. Specify other options for the query such as output columns, search conditions, and sort order.

[Show All](#)

Enforce referential integrity between tables (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Referential integrity between tables is enforced by default when you create a relationship in your database diagram. An enforced relationship ensures each value entered in a foreign key column matches an existing value in the related primary key column.

You can change the conditions under which referential integrity is enforced by editing the relationship's properties.

[Change referential integrity options for a new relationship](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, create a relationship.

[How?](#)

You create a relationship between two tables when you want to associate rows of one table with rows of another.

[Create a relationship in a database diagram](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector  for the database column or combination of columns that you want to relate to a column in another table.
3. While the pointer is positioned over the row selector, click and drag to

- the related table.
4. Release the mouse button. The **Create Relationship** dialog box appears and attempts to match the columns you selected with columns of the same name and data type in the related table.
 5. In the **Create Relationship** dialog box, confirm that the columns you want to relate are shown in the **Primary key table** and **Foreign key table** lists.
 6. Choose **OK** to create the relationship.

On the diagram, the primary key side of the relationship is denoted by a key  symbol. In one-to-one relationships, the table that initiated the relationship determines the primary key side. For example, if you create a relationship from the pub_id column in the publishers table to the pub_id column in the pub_info table, then the publishers table is on the primary key side of the relationship.

[Create a relationship in Table Designer](#)

1. Click **Tables**  under **Objects**, click the table that is on the foreign key side of the relationship, and then click **Design** on the Database window toolbar.
 2. Right-click in the Table Designer and choose **Relationships**.
 3. Click the **New** button.
 4. From the drop-down list in **Primary Key Table**, choose the table that will be on the primary-key side of the relationship. In the grid beneath, enter the columns contributing to the table's primary key. In the adjacent grid cell to the left of each column, enter the corresponding foreign-key column of the foreign-key table.
The table designer suggests a name for the relationship. To change this name, edit the contents of the **Relationship Name** text box.
 5. Choose **Close** to create the relationship.
3. In the **Create Relationship** dialog box, clear or select one or more of the options. For more information about each option, click **Help**.

[Change referential integrity options for an existing relationship](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the

Database window toolbar.

2. In your database diagram, select the relationship line.
3. Right-click the relationship line and select **Properties**.
4. Choose the **Relationships** tab.
5. Select the relationship from the **Selected relationship** list.
6. Clear or select one or more of the options. For more information about each option, click **Help**.

The relationship is updated in the database when you save the database diagram or either of the related tables.

Draw a reflexive relationship (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You create a reflexive relationship to link a column or columns in a table with another column or columns in the same table. For example, suppose the `employee` table has an `emp_id` column and a `mgr_id` column. Because each manager is also an employee, you relate these two columns by drawing a relationship line from the table to itself. This relationship ensures each manager ID that is added to the table matches an existing employee ID.

Before you create a relationship, you must first define a primary key or unique constraint for your table. You then relate the primary key column to a matching column. Once you create the relationship, the matching column becomes a foreign key of the table.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector for the database column or columns that you want to relate to another column or columns.
3. While the pointer is positioned over the row selector, drag the pointer outside the table until a line appears.
4. Drag the line back to the selected table.
5. Release the mouse button. The **Create Relationship** dialog box appears and attempts to match the primary key columns with the nonkey columns you dragged the line to.
6. Confirm that the columns you want to relate are shown in the **Primary key table** and **Foreign key table** lists.
7. Choose **OK** to create the relationship.

When you run queries against a table, you can use a reflexive relationship to create a self-join.

Rename a relationship (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can rename a relationship. When you rename a relationship, the relationship name is automatically updated in the current database diagram and any other database diagram in which it appears.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the line representing the relationship that you want to rename.
3. Right-click the relationship line and select **Properties**.
4. Choose the **Relationships** tab.
5. In the **Relationship name** box, type a new name.

The relationship is renamed in the database diagram and any other database diagram in which it appears as soon as you exit the **Relationship name** box; it is renamed in the database when you save either of the related tables or the database diagram.

[Show All](#)

Map many-to-many relationships in a database diagram (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

Many-to-many relationships let you relate each row in one table to many rows in another table, and vice versa. For example, you could create a many-to-many relationship between the `authors` table and the `titles` table to match each author to all of his or her books and to match each book to all of its authors. Creating a one-to-many relationship from either table would incorrectly indicate that every book can have only one author, or that every author can write only one book.

Many-to-many relationships between tables are accommodated in databases by means of junction tables. A junction table contains the primary key columns of the two tables you want to relate. You then create a relationship from the primary key columns of each of those two tables to the matching columns in the junction table. In the Microsoft SQL Server `pubs` sample database, the `titleauthor` table is a junction table.

[Create a many-to-many relationship between tables](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, add the tables that you want to create a many-to-many relationship between.

[How?](#)

You can add a table to your database diagram to edit its structure or relate it to other tables in your diagram. You can either add existing database tables

to a database diagram or insert a new table that has not yet been defined in the database. Alternatively, you can create a table or modify an existing table with the Table Designer.

 [Insert a new table in a database diagram](#)

Adding a new table to the database diagram means that you are defining a new table that does not already exist in the database. To create a new table, you must define the individual columns that make up the table. The table is created in the database when you save the table or the database diagram.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.

To create a table in your current diagram, click **New Table** on the **Insert** menu.

–or–

Right-click in the diagram and select **New Table**.

2. To create a table in your current diagram, click **New Table** on the **Insert** menu.

–or–

Right-click in the diagram and select **New Table**.

3. Modify or accept the system-assigned table name, in the **Choose Name** dialog box, and then choose **OK**.
A new table appears in the diagram in column properties view.
4. In the first cell of the new table, type a column name. Then press the TAB key to move to the next cell.
5. Under **Data Type**, select a data type for the column. Each column must have a name and data type.
You can set the column's other properties, such as Scale, by using the **Columns** property page.
6. Repeat steps 3 and 4 for each column you want to add to the table.

When you save your database diagram, the new table will be added to your database.

Note If you create a new table, then remove it from the database diagram before saving it to the database, the table name remains in memory until you close the database diagram. Also, if you delete an existing table from the database, the table name remains in memory. To use the table name again, close and restart the Database Designer.

 [Add existing tables to a database diagram](#)

Adding an existing table means that the table you want to appear in your diagram already exists in your database. If you add a group of tables to a diagram, any relationships that exist between the tables are also added to the diagram.

1. Right click on the database diagram, and then click **Add Table**
 .
2. Select the table you want in the **Tables** list, and then click **Add**.
3. Repeat step 2 for each additional table you want to add.
4. Click **Close**.

If relationships exist between the selected table and other tables in your database diagram, relationship lines are automatically drawn.

When you add a table to a database diagram, the definition of the table (not the data that is stored in the table) is loaded from your database into memory. At that point you can edit the table's definition. For example, you can add new columns or modify its indexes, key, relationships, or constraints.

 [Add related tables to a database diagram](#)

For tables with existing foreign key constraints, you can easily add the related tables to the database diagram.

1. Select one or more tables with foreign key constraints in the database diagram.
2. Right-click on any of the selected tables and choose **Add Related Tables**.

Both those tables referenced by a foreign key constraint from the selected table(s) and those referencing the selected table(s) with a foreign key constraint are added to the database diagram.

3. Create a third table by right-clicking the diagram and choosing **New Table** from the shortcut menu. This will become the junction table.
4. In the **Choose Name** dialog box, change the system-assigned table name. For example, the junction table between the `titles` table and the `authors` table is now named `titleauthors`.
5. Copy the primary key columns from each of the other two tables to the junction table. You can add other columns to this table, just as you can to any other table.

 [How?](#)

You can copy columns from one table to another table in the same database diagram or in different database diagrams. Copying a column involves only the column definition. The data itself is not automatically transferred to the second table as part of this process.

 [Tip](#)

You can use queries to copy the column's data from the original column to the new column. Use the Query Designer to run an update query to add the data to the table to which you copied the columns.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the columns that you want to copy.
3. Click the **Copy** button on the toolbar. This action copies the selection from the table and places the column and its current set of properties on the Clipboard.
4. Position the cursor in the table at the location where you want to insert

the columns.

5. Click the **Paste** button on the toolbar. The column and its properties are inserted at the new location.

When you copy a database column that has a user-defined data type from one database to another, the user-defined data type may not be available in the destination database. In such a case, the column will be assigned the nearest matching base data type available in that database.

6. In the junction table, set the primary key to include all the primary key columns from the other two tables.

 [How?](#)

Define a primary key to enforce uniqueness for values entered in specified columns that do not allow nulls. If you define a primary key for a table in your database, you can relate that table to other tables, thus reducing the need for redundant data. A table can have only one primary key.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector for the database column you want to define as the primary key. If you want to select multiple columns, hold down the CTRL key while you click the row selectors for the other columns.
3. Right-click the row selector for the column and select **Primary Key**. A primary key index, named "PK_" followed by the table name, is automatically created; you can find it on the **Indexes/Keys** tab of the **Properties** page.

Warning If you want to redefine the primary key, any relationships to the existing primary key must be deleted before the new primary key can be created. A message will warn you that existing relationships will be automatically deleted as part of this process.

A primary key column is identified by a primary key symbol  in its row selector.

If a primary key consists of more than one column, duplicate values are allowed in one column, but each combination of values from all the columns in the primary key must be unique.

If you define a compound key, the order of columns in the primary key matches the order of columns as shown in the table in your database diagram. However, you can change the order of columns after the primary key is created. In the **Column name** grid, remove the columns from the primary key. Then add the columns back in the order you want. To remove a column from the key, simply remove the column name from the Column name list.

7. Define a one-to-many relationship between each of the two primary tables and the junction table. The junction table should be at the “many” side of both of the relationships you create.

[How?](#)

You create a relationship between two tables when you want to associate rows of one table with rows of another.

[Create a relationship in a database diagram](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector  for the database column or combination of columns that you want to relate to a column in another table.
3. While the pointer is positioned over the row selector, click and drag to the related table.
4. Release the mouse button. The **Create Relationship** dialog box appears and attempts to match the columns you selected with columns of the same name and data type in the related table.
5. In the **Create Relationship** dialog box, confirm that the columns you want to relate are shown in the **Primary key table** and **Foreign key table** lists.
6. Choose **OK** to create the relationship.

On the diagram, the primary key side of the relationship is denoted by a key symbol. In one-to-one relationships, the table that initiated the relationship determines the primary key side. For example, if you create a relationship from the `pub_id` column in the `publishers` table to the `pub_id` column in the `pub_info` table, then the `publishers` table is on the primary key side of the relationship.

[Create a relationship in Table Designer](#)

1. Click **Tables**  under **Objects**, click the table that is on the foreign key side of the relationship, and then click **Design** on the Database window toolbar.
2. Right-click in the Table Designer and choose **Relationships**.
3. Click the **New** button.
4. From the drop-down list in **Primary key table**, choose the table that will be on the primary-key side of the relationship. In the grid beneath, enter the columns contributing to the table's primary key. In the adjacent grid cell to the left of each column, enter the corresponding foreign-key column of the foreign-key table.
The table designer suggests a name for the relationship. To change this name, edit the contents of the **Relationship name** text box.
5. Choose **Close** to create the relationship.

Note The creation of a junction table in a database diagram does not insert data from the related tables into the junction table.

Modify a join operator in a query (ADP)

By default, the Query Designer joins tables using an equal sign (an equijoin), which matches values in the two join columns. If you want, you can change the operator used to compare values in the join columns.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the join line you want to modify, and then choose **Properties** from the shortcut menu.
3. In the **Join Line** tab of the **Properties** dialog box, select a new operator from the list.

Note You can also change the operator in the SQL statement in the SQL pane.

[Show All](#)

Create a join between tables on multiple columns in a query (ADP)

You can join tables with multiple columns. That is, you can create a query that matches rows from the two tables only if they satisfy multiple conditions. If the database contains a relationship matching multiple foreign-key columns in one table to a multicolumn primary key in the other table, you can use this relationship to create a multicolumn join.

Even if the database contains no multi-column foreign-key relationship, you can create the join manually.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add to the Diagram pane the tables you want to join.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane

accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.



[Add an existing query to a new query](#)

1. If necessary, click **SQL**  to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Drag the name of the first join column in the first table window and drop it onto the related column in the second table window. You cannot base a join on text, ntext, or image columns.

Note In general, the join columns must be of the same (or compatible) data types. For example, if the join column in the first table is a date, you must relate it to a date column in the second table. On the other hand, if the first join column is an integer, the related join column must also be of an

integer data type, but it can be a different size. However, Microsoft SQL Server provides implicit data type conversions so that many joins between seemingly incompatible columns will work.

The Query Designer will not check the data types of the columns you use to create a join, but when you execute the query, the database will display an error if the data types are not compatible.

4. Drag the name of the second join column in the first table window and drop it onto the related column in the second table window.

[Show All](#)

Create a join between tables, views, or functions in a query manually (ADP)

When you add two (or more) tables to a query, the Query Designer attempts to join them based on common data or on information stored in the database about how tables are related. However, if the Query Designer has not joined the tables automatically, or if you want to create additional join conditions between tables, you can join tables manually.

You can create joins based on comparisons between any two columns, not just columns that contain the same information. For example, if your database contains two tables, `titles` and `roysched`, you can compare values in the `ytd_sales` column of the `titles` table against the `lorange` and `hirange` columns in the `roysched` table. Creating this join would allow you to find titles for which the year-to-date sales falls between the low and high ranges for the royalty payments.

Tip

Joins work fastest if the columns in the join condition have been indexed. In some cases, joining on unindexed columns can result in a slow query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add to the Diagram pane the objects you want to join.

How?

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.



[Add an existing query to a new query](#)

1. If necessary, click **SQL**  to show the **SQL** pane.
2. In the SQL pane, type a right and left parentheses () after the word FROM.
3. Open the Query designer for the existing query . (You now have two Query Designers open.)
4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
5. Select all the text in the SQL pane, and copy it to the Clipboard.
6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.

7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Drag the name of the join column in the first table, view, or function and drop it onto the related column in the second table, view, or function. You cannot base a join on text, ntext, or image columns.

Note The join columns must be of the same (or compatible) data types. For example, if the join column in the first table is a date, you must relate it to a date column in the second table. On the other hand, if the first join column is an integer, the related join column must also be of an integer data type, but it can be a different size. The Query Designer will not check the data types of the columns you use to create a join, but when you execute the query, the database will display an error if the data types are not compatible.

4. If necessary, change the join operator; by default, the operator is an equal sign (=).



By default, the Query Designer joins tables using an equal sign (an equijoin), which matches values in the two join columns. If you want, you can change the operator used to compare values in the join columns.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the join line you want to modify, and then choose **Properties** from the shortcut menu.
3. In the **Join Line** tab of the **Properties** dialog box, select a new operator from the list.

Note You can also change the operator in the SQL statement in the SQL pane.

The Query Designer adds an INNER JOIN clause to the SQL statement in the SQL pane.

Remove a join in a query (ADP)

If you do not want tables to be joined via an inner join or an outer join, you can remove the join between them. For example, you might remove a join that the Query Designer has created automatically between two tables.

Note Removing a join from a query does alter the underlying relationship in the database.

If two joined tables are part of your query and you remove all join conditions between them, the resulting query becomes the product of both tables— that is, it becomes a CROSS JOIN (also called a Cartesian product).

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, select the join line for the join to remove, and then press the DELETE key. You can select and delete multiple join lines at one time by using the SHIFT key.

The Query Designer removes the join line and alters the statement in the SQL pane.

Delete a relationship (ADP)

Delete a relationship when you no longer want to relate columns in two related tables. When you redesign tables, it is often necessary to delete relationships and then recreate them after your new design is complete. For example, if you decide to normalize a database and store all address data in one table, you would delete all the relationships to address columns between existing tables, create a new table containing the address columns, and then create relationships from the new address table to every table that requires an address.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, select the line that represents the relationship that you want to delete from the diagram.
3. Right-click the relationship line and choose **Delete Relationship from Database** from the shortcut menu.
4. A message box prompts you to confirm the deletion. Choose **Yes**.

Note When you delete a relationship, the relationship line is removed from every diagram in which it appears. It is deleted from the database when you save the database diagram or when you save either of the tables that it related.

[Show All](#)

Troubleshoot queries (ADP)

[I'm having problems creating a query.](#)

The following are common problems you may encounter when creating a query:

- The syntax you entered is valid but is not supported visually by the Query Designer. Be sure to verify your syntax before saving by using the **Verify SQL Syntax** button .
- Expressions with different data types can generate errors. To work with different data types in a single expression, you need to convert to the same data type using either the CONVERT or CAST function. For more information on these functions, see the SQL Server documentation.
- Make sure the number of items in the Column list in the UPDATE and INSERT INTO statements matches the number of items in the Values list. Verify your SQL statement using the **Verify SQL Syntax** button  to make sure the number of items match.
- A function in your query has the wrong number of arguments. Review your query syntax to locate the function and provide the correct number of arguments. In a user-defined function, you must always enter parameter arguments if there are arguments, and you must always enter parentheses, even if there are no arguments.
- When you use user-defined functions, SQL Server does not automatically promote other integer data types (such as **tinyint**, **smallint**, and **int**) to **bigint**; SQL Server only returns **bigint** if the parameter expression is a **bigint** data type.
- You may be trying to pass an expression as a parameter value. Parameter values must be scalar values or simple text.
- When specifying a criteria for the EXISTS statement, you must use either True for EXISTS or False for NOT EXISTS.
- You cannot use The NOT keyword in a column cell. When entering a NOT EXIST subquery into the Column cell, enter just the EXIST clause. Then go to the **Criteria** cell and enter False for it.
- An alias you supplied contains embedded blanks or other white space. Remove the white space or put double-quotation marks around the alias.

[I get the message "The table-valued function used as a target is not an in-line function."](#)

The operation you are attempting requires an updateable user-defined function (an in-line function), but the function you are using is not updateable. You can replace the function with an updateable one, or you can change the query to a Select query.

[I'm having problems creating or modifying indexed views.](#)

You cannot use the HAVING clause in the view definition of an indexed view. Either remove the indexes or remove the HAVING clause.

You cannot use the asterisk (*) **All Columns** check box to select all columns in an indexed view. Because an indexed view is stored on disk, the view definition must explicitly refer to the columns to be included in the view. Replace the asterisk (*) with the columns you want to include in the view.

An indexed view must contain a clustered index because it is stored on disk. Modify the index to make it a clustered index.

If you remove schema binding, the indexes will be dropped. Some operations, such as renaming a base table column or deleting a base table, automatically remove schema binding from an indexed view. When schema binding is removed, the indexes will be dropped.

Note Indexed views are supported when your Microsoft Access project is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

[I'm seeing unexpected results with my query.](#)

You might see unexpected results when you use a SQL statement whose output contains two or more fields with the same name. For example, you might see

unexpected behavior when you work with a view that is based on two tables and that contains two fields that have the same name but are located in different tables. To prevent unexpected behavior, specify an alias for one of the fields to ensure that every field in the output has a unique name.

 [Access added a CONVERT and DATETIME function to my date expression.](#)

Access adds a CONVERT Transact-SQL function to convert any date expressions in your SQL statements to an explicit four-digit year format as the following example shows:

```
CONVERT(DATETIME, '1997-01-01 00:00:00', 102)
```

For more information on the CONVERT Transact-SQL function, see the SQL Server documentation.

 [My Make-Table query fails to run.](#)

A Make-Table query uses the INSERT FROM SQL Statement to copy rows from one table to another. This situation might occur when you upsize your Access database from a Microsoft SQL Server version 7.0 or 6.5 database, create your own make-table stored procedure in the SQL Text Editor, or create a Make-Table query using the SQL Server Enterprise Manager in SQL Server.

To run a Make-Table query in an Access project connected to a Microsoft SQL Server version 7.0 or 6.5 database, you must set the select into/bulkcopy database option to true (the default value is false) using the system stored procedure sp_dboption as the following example shows.

```
Exec sp_dboption 'select into/bulkcopy'
```

You can run the system stored procedure in SQL Server Enterprise Manager or create a stored procedure in Access using the SQL Text Editor. To set this option, you must be a member of the db_owner or sysadmin roles. For more information on the select into/bulkcopy database option, see the SQL Server documentation.

 [I get the message "ADO Error: Invalid Use of <function name> Within a](#)

Function."

You cannot use nondeterministic Transact-SQL functions, nondeterministic global variables, or extended stored procedures in a CREATE FUNCTION SQL statement. Nondeterministic functions, global variables, and extended stored procedures can return different result values each time they are called with the same set of input values, and therefore have possible side-effects in the database, such as table updates.

The following functions and global variables are not allowed in a user-defined function:

```
@@CONNECTIONS
@@CPU_BUSY
@@IDLE
@@IO_BUSY
@@MAX_CONNECTIONS
@@PACK_RECEIVED
@@PACK_SENT
@@PACKET_ERRORS
@@TIMETICKS
@@TOTAL_ERRORS
@@TOTAL_READ
@@TOTAL_WRITE
GETDATE
GETUTCDATE
NEWID
RAND
TEXTPTR
```

[When I run a named parameter query, I get an unexpected blank result set.](#)

You may be using words in the parameter prompt that are not supported. You can't use Visual Basic for Application (VBA) language keywords that are part of the Access object model as parameter names. To see a list of these language keywords, use the Object Browser in the Visual Basic Editor to display the members of the Access Application object.

[I'm having problems using domain functions in my parameter query.](#)

You cannot use the following domain aggregate functions in a Microsoft Access project (.adp).

DAvg
DCount
DLookup
DFirst, DLast
DMin, DMax
DStDev, DStDevP
DSum
DVar, DVarP



[I'm having problems using domain aggregate functions in a form or report.](#)

In an Access project, your aggregate functions cannot use references to controls in objects such as forms or reports. For example, you cannot use the following expression:

```
=DLookup("CategoryName";"Categories";"CategoryID = Reports!Report1!C
```

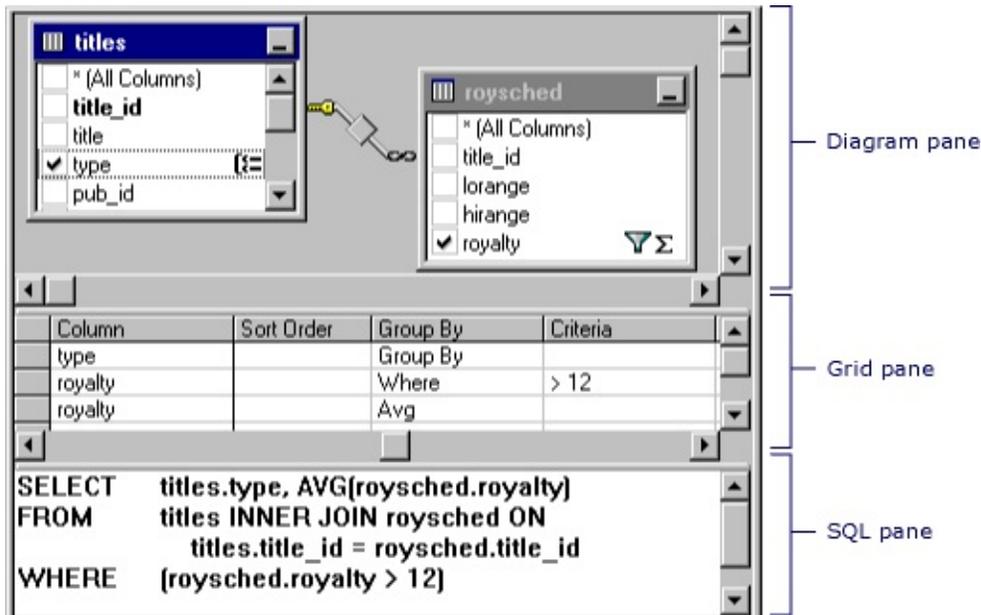
[Show All](#)

About the Query Designer layout (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The Query Designer consists of three panes: the Diagram pane, the Grid pane, and the SQL pane.

[The Three Panes of the Query Designer](#)



- **The Diagram pane** displays the tables, views, and in-line functions that you are querying. Each rectangle represents a table, view, or in-line function and shows the available data columns as well as icons that indicate how each column is used in the query. Joins are indicated by lines between the rectangles.
- **The Grid pane** contains a spreadsheet-like grid in which you specify options, such as which data columns to display, what rows to select, how to group rows, and so on.
- **The SQL pane** displays the SQL statement for the query. You can edit the

SQL statement created by the Query Designer or you can enter your own SQL statement. It is particularly useful for entering SQL statements that cannot be created using the Diagram and Grid panes, such as union queries.

You can create a query by working in any of the panes: you can specify a column to display by choosing it in the Diagram pane, entering it into the Grid pane, or making it part of the SQL statement in the SQL pane. The Diagram, Grid, and SQL panes are synchronized— when you make a change in one pane, the other panes automatically reflect the change.

[The Diagram Pane](#)

[Overview of the Diagram pane](#)

The Diagram pane presents a graphic display of the tables, views, user-defined functions, or subqueries in your query. It also shows any join relationships among them.

In the Diagram pane you can:

- Add or remove tables, views, user-defined functions, or subqueries and specify data columns for output.
- Specify columns for ordering the query.
- Specify that you want to group rows in the result set.
- Create or modify joins between tables, views, user-defined functions, or subqueries.

When you make a change in the Diagram pane, the Grid pane and SQL pane are updated to reflect your change. For example, if you select a column for output in a table, view, user-defined function, or subquery window in the Diagram pane, the Query Designer adds the data column to the Grid pane and to the SQL statement in the SQL pane.

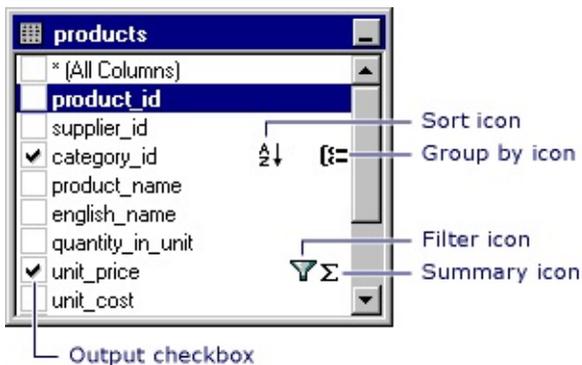
[About the Diagram pane icons, check boxes, and symbols](#)

Each table, view, user-defined function, or subquery appears as a separate window in the Diagram pane. The icon in the title bar of each rectangle indicates what type of object the rectangle represents, as illustrated in the following table.

Icon Object type

-  Table
-  View
-  In-line Function
-  Subquery (in FROM clause)

Each rectangle shows the data columns for the table, view, or user-defined function. Check boxes and symbols appear next to the names of columns to indicate how the columns are used in the query. ToolTips display information such as data type and size for columns.



The following table lists the check boxes and symbols used in the rectangle for each table, view, or user-defined function.

Check box or symbol

Description

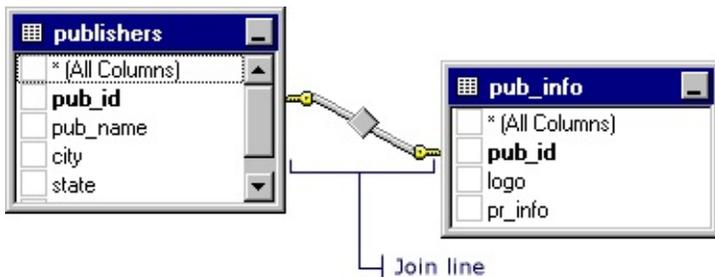
- Specifies whether a data column appears in the query result set (Select query) or is used in an Update, Append, Make- Table, or Append Values query. Select the column to add it to the results. If **(All Columns)** is selected, all data columns appear in the output.
-  The icon used with the check box changes according to the type of query you are creating. When creating a Delete query, you cannot select individual columns.
-  Indicates that the data column is being used to order the query results (is part of an ORDER BY clause). The icon appears as A-Z if the sort order is ascending or Z-A if sort order is descending.
-  Indicates that the data column is being used to create a grouped result

- ⊆ set (is part of a GROUP BY clause) in an aggregate query.
- ∇ Indicates that the data column is included in a search condition for the query (is part of a WHERE or HAVING clause).
- Σ Indicates that the contents of the data column are being summarized for output (are included in a SUM, AVG, or other aggregate function).

Note The Query Designer will not display data columns for a table, view, or user-defined function if you do not have sufficient access rights to it or if the database driver cannot return information about it. In such cases, the Query Designer displays only a title bar for the table, view, or user-defined function.

 [Joined tables on the Diagram pane](#)

If the query involves a join, a join line appears between the data columns involved in the join. If the joined data columns are not displayed (for example, the table, view, user-defined function, or subquery window is minimized or the join involves an expression), the Query Designer places the join line in the title bar of the rectangle representing the table, view, user-defined function, or subquery. The Query Designer displays one join line for each join condition.



The shape of the icon in the middle of the join line indicates how the tables or table-structured objects are joined. If the join clause uses an operator other than equal (=), the operator is displayed in the join line icon. The following table lists the icons that can be displayed in a join line.

Join line icon	Description
	Inner join (created using equal sign).
	Inner join based on the "greater than" operator. (The operator displayed

-  in the join line icon reflects the operator used in the join.)
-  Outer join in which all rows from the table represented on the left will be included, even if they do not have matches in the related table.
-  Outer join in which all rows from the table represented on the right will be included, even if they do not have matches in the related table.
-  A full outer join in which all rows from both tables will be included, even if they do not have matches in the related table, view, user-defined function, or subquery.

Icons on the ends of the join line indicate the type of join. The following table lists the types of joins and the icons that can be displayed on the ends of the join line.

Icon on ends of join line	Description
	One-to-one join
	One-to-many join
	Query Designer cannot determine join type

 [The Grid Pane](#)

 [Overview of the Grid pane](#)

The Grid pane allows you to specify query options— such as which data columns to display, how to order the results, and what rows to select— by entering your choices into a spreadsheet-like grid. In the Grid pane you can specify:

- Columns to display and column name aliases.
- The table that a column belongs to.
- Expressions for calculated columns.
- The sort order for the query.
- Search conditions.
- Grouping criteria, including aggregate functions to use for summary reports.
- New values for Update or Append Values queries.

- Target column names for Append queries.

Changes you make in the Grid pane are automatically reflected in the Diagram pane and SQL pane. Similarly, the Grid pane is updated automatically to reflect changes made in the other panes.

[About the Grid pane columns](#)

The rows in the Grid pane display the data columns used in your query; columns in the Grid pane display query options.

Query options appear in grid columns

Column	Output	Sort Type	Sort Order	Criteria	Or...
employee_id	<input checked="" type="checkbox"/>				
last_name	<input checked="" type="checkbox"/>	Ascending	1		
first_name	<input checked="" type="checkbox"/>				
birth_date	<input checked="" type="checkbox"/>	Ascending	2	> "01 Dec 26"	

Data columns appear as rows in the grid

The specific information that appears in the Grid pane depends on the type of query you are creating. If you are creating a Select query, the Grid pane contains different columns than if you are creating an Update query.

The following table lists the grid columns that can appear in the Grid pane.

Column	Query type	Description
Column	All	Displays either the name of a data column used for the query or the expression for a computed column. This column is locked so that it is always visible as you scroll horizontally.
Alias	Select, Append, Update, Make-Table	Specifies either an alternate name for a column or the name you can use for a computed column.
Table	Select, Append, Update, Make-	Specifies the name of the table, view, user-defined function, or subquery for the associated data column. This column is blank for computed columns.

	Table	
Output	Select, Append, Make-Table	Specifies whether a data column appears in the query output. Note If the database allows, you can use a data column for sort or search clauses without displaying it in the result set.
Sort Type	Select, Append	Specifies that the associated data column is used to sort the query results and whether the sort is ascending or descending.
Sort Order	Select, Append	Specifies the sort priority for data columns used to sort the result set. When you change the sort order for a data column, the sort order for all other columns is updated accordingly. Specifies that the associated data column is being used to create an aggregate query. This grid column appears only if you have chosen Group By from the Tools menu or have added a GROUP BY clause to the SQL pane.
Group By	Select, Append, Make-Table	By default, the value of this column is set to Group By , and the column becomes part of the GROUP BY clause. When you move to a cell in this column and select an aggregate function to apply to the associated data column, by default the resulting expression is added as an output column for the result set.
		Specifies a search condition (filter) for the associated data column. Enter an operator (the default is "=") and the value to search for. Enclose text values in single quotation marks. If the associated data column is part of a GROUP BY clause, the expression you enter is used for a HAVING clause.
Criteria All		If you enter values for more than one cell in the Criteria grid column, the resulting search conditions are automatically linked with a logical AND. To specify multiple search condition expressions for a single database column (for example, (fname > 'A') AND (fname < 'M')), add the data column to the Grid pane twice and enter

separate values in the **Criteria** grid column for each instance of the data column.

Or ... All Specifies an additional search condition expression for the data column, linked to previous expressions with a logical OR. You can add more **Or ...** grid columns by pressing the TAB key in the rightmost **Or ...** column.

Append Append Specifies the name of the target data column for the associated data column. When you create an Append query, the Query Designer attempts to match the source to an appropriate target data column. If the Query Designer cannot choose a match, you must provide the column name.

New Value Update, Append Values Specifies the value to place into the associated column. Enter a literal value or an expression.

 [The SQL Pane](#)

 [Overview of SQL pane](#)

The SQL pane displays the SQL statement for the current query. As you build your query, the SQL pane automatically updates and reformats to be easy to read.

In the SQL pane you can:

- Create new queries by entering SQL statements.
- Modify the SQL statement created by the Query Designer based on settings you make in the Diagram and Grid panes.
- Enter statements that take advantage of features specific to Microsoft SQL Server.

 [Statements in the SQL pane](#)

You can edit the current query directly in the SQL pane. When you move to another pane, the Query Designer automatically formats your statement, and then changes the Diagram and Grid panes to match your statement.

Note You can enter optimizer hints for SQL statements, but the Query Designer might reformat them.

If your statement cannot be represented in the Diagram and Grid panes, and if those panes are visible, the Query Designer displays an error and then offers you two choices:

- Return to the SQL pane and edit the statement.
- Discard your changes and revert to the most recent version of the SQL statement.

If you return to the SQL pane and continue editing the statement, the Query Designer dims the other panes to indicate that they no longer reflect the contents of the SQL pane.

You can also use the SQL pane to enter SQL statements that cannot be represented graphically in the Query Designer. In such cases, the Query Designer displays the same behavior as it does when it detects an error— it dims the Diagram and Grid panes to indicate that they do not represent the current statement. You can continue to edit the statement and execute it as you would any SQL statement.

Note If you enter an SQL statement, but then make further changes to the query by changing the Diagram and Grid panes, the Query Designer rebuilds and redisplay the SQL statement. In some cases, this action results in an SQL statement that is constructed differently from the one you originally entered (though it will always yield the same results). This difference is particularly likely when you are working with search conditions that involve several clauses linked with AND and OR.

[Show All](#)

About summary and grouping behavior in the Query Designer (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you create aggregate queries, certain logical principles apply. For example, you cannot display the contents of individual rows in a summary query. The Query Designer helps you comply with these principles in the way the Diagram and Grid panes behave.

By understanding the principles of aggregate queries and the Query Designer's behavior, you can create logically correct aggregate queries. The overriding principle is that aggregate queries can result only in summary information. Thus, most of the principles that follow describe the ways that you can reference individual data columns within an aggregate query.

[Referencing columns for output and sorting](#)

The following principles describe how you can reference columns in an aggregate query for output and for sorting:

- If you include an aggregate function anywhere in a query, the query is considered an aggregate query. (This principle does not necessarily apply to subqueries, which can include nonaggregate information.)
- You cannot display the contents of individual rows in an aggregate query; you can display only summary data. As a consequence, all columns marked for output must also be assigned to either an aggregate function or to the GROUP BY clause.
- Columns used in aggregate functions cannot appear in the GROUP BY clause.

[Referencing columns in search conditions](#)

The following principles describe how you can reference columns in an aggregate query in search conditions. In general, you can use a column in a search condition to filter the rows that should be summarized (a WHERE clause) or to determine which grouped results appear in the final output (a HAVING clause).

- Individual data columns can appear in either the WHERE or HAVING clause, depending on how they are used elsewhere in the query.
- WHERE clauses are used to select a subset of rows for summarizing and grouping and are thus applied before any grouping is done. Therefore, you can use a data column in a WHERE clause even if it is not part of the GROUP BY clause or contained in an aggregate function. For example, the following statement selects all titles that cost more than \$10.00 and averages the price:

```
SELECT AVG(price)
FROM titles
WHERE price > 10
```

- If you create a search condition that involves a column also used in a GROUP BY clause or aggregate function, the search condition can appear as either a WHERE clause or a HAVING clause—you can decide which when you create the condition. For example, the following statement creates an average price for the titles for each publisher, then displays the average for the publishers in which the average price is greater than \$10.00:

```
SELECT pub_id, AVG(price)
FROM titles
GROUP BY pub_id
HAVING (AVG(price) > 10)
```

- If you use an aggregate function in a search condition, the condition involves a summary and must therefore be part of the HAVING clause.

[Working with columns in aggregate queries](#)

When you create aggregate queries using the Diagram and Grid panes, the Query Designer makes certain assumptions so that your query adheres to the principles outlined earlier. For example, if you are creating an aggregate query and mark a data column for output, the Query Designer automatically makes the column part

of the GROUP BY clause so that you do not inadvertently attempt to display the contents of an individual row in a summary.

Even though the Query Designer works to help prevent logical errors, it is possible to create aggregate queries that will not execute. Therefore, be sure that you understand the principles listed earlier in order to avoid errors when you create queries.

The Query Designer uses the following guidelines for working with columns:

- When you choose the Group By option or add an aggregate function to a query, all columns marked for output or used for sorting are automatically added to the GROUP BY clause. Columns are not automatically added to the GROUP BY clause if they are already part of an aggregate function.

If you do not want a particular column to be part of the GROUP BY clause, you must manually change it by selecting a different option in the Group By column of the Grid pane. However, the Query Designer will not prevent you from choosing an option that can result in a query that will not run.

- If you manually add a query output column to an aggregate function in either the Grid or SQL pane, the Query Designer does not automatically remove other output columns from the query. Therefore, you must remove the remaining columns from the query output or make them part of the GROUP BY clause or of an aggregate function.

When you enter a search condition into the Criteria column of the Grid pane, the Query Designer follows these rules:

- If the **Group By** column of the grid is not displayed (because you have not yet specified an aggregate query), the search condition is placed into the WHERE clause.
- If you are already in an aggregate query and have selected the option **Where** in the **Group By** column, the search condition is placed into the WHERE clause.
- If the **Group By** column contains any value other than **Where**, the search condition is placed in the HAVING clause.

[Show All](#)

About using the Query Designer with international data (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can use the Query Designer with data in any language and in any version of the Windows operating system. The following guidelines outline the differences you will notice and provide information about managing data in international applications.



[Localized information in the Grid and SQL panes](#)

If you are using the Grid pane to create your query, you can enter information in the format that corresponds to the Windows Regional Settings for your computer. For example, if you are searching for data, you can enter the data in the Criteria columns using whatever format you are accustomed to using, with these exceptions:

- Long data formats are not supported.
- Currency symbols should not be entered in the Grid pane. However, the correct symbol will appear in datasheet view.

Note You may enter the \$ currency symbol, but the Results pane will return the currency symbol that corresponds to the Windows Regional Settings for your computer.

- Unary minus always appears on the left side (for example, -1) regardless of the Regional Settings options.

In contrast, data and keywords in the SQL pane must always be in ANSI (U.S.) format. For example, as the Query Designer builds a query, it inserts the ANSI form of all SQL keywords such as SELECT and FROM. If you add elements to the statement in the SQL pane, be sure to use the ANSI standard form for the elements.

When you enter data using local-specific format in the Grid pane, the Query Designer automatically translates it to ANSI format in the SQL pane. For example, if your Regional Settings are set to Standard German, you can enter data in the Grid pane in a format such as “31.12.96.” However, the date will appear in the SQL pane in ANSI datetime format as ‘1996-12-31 00:00:00’. If you enter data directly in the SQL pane, you must enter it in ANSI format.

[Sort order](#)

The sort order of data in your query is determined by the database. Options that you set in the **Windows Regional Settings** dialog box do not affect sort order for queries. Within any particular query, however, you can request that rows be returned in a particular order. For more information on sort order and collations, see the Microsoft SQL Server documentation.

[Using double-byte characters](#)

A double-byte character set (DBCS) uses two bytes to represent a character, allowing more than 256 characters to be represented. A DBCS is typically used in environments that use ideographic writing systems, such as Japanese, Korean, and Chinese. You can enter DBCS characters for literals and for database object names such as table and view names or aliases. You can also use DBCS characters for parameter names and parameter marker characters. However, you cannot use DBCS characters in SQL language elements such as function names or SQL keywords.

[Show All](#)

About creating queries (ADP)

A query requests data from the database. At its simplest, a query merely fetches all data from a single table. But as you create more complex (and more typical) queries, you can assemble exactly the data you want in the following ways:

[Including or excluding rows](#)

[Using search conditions](#)

To restrict the number of rows a query should return, you create search conditions or filter criteria. In SQL, search conditions appear in the WHERE clause of the statement, or if you are creating an aggregate query, in the HAVING clause.

Note You can also use search conditions to indicate which rows are affected by an Update, Update Values, Delete, or Make-Table query.

When the query runs, the database engine examines and applies the search condition to each row in the tables you are searching. If the row meets the condition, it is included in the query. For example, a search condition that would find all the employees in a particular region might be:

```
region = 'UK'
```

To establish the criteria for including a row in a result, you can use multiple search conditions. For example, the following search criterion consists of two search conditions. The query includes a row in the result set only if that row satisfies both of the conditions.

```
region = 'UK' AND product_line = 'Housewares'
```

You can combine these conditions with AND or OR. The previous example uses AND. In contrast, the following criterion uses OR. The attendant query result will include any row that satisfies either or both of the search conditions:

```
region = 'UK' OR product_line = 'Housewares'
```

You can even combine search conditions on a single column. For example, the following criterion combines two conditions on the region column:

```
region = 'UK' OR region = 'US'
```



[Predicates in search conditions](#)

A search condition consists of one or more predicates, each specifying a single condition. If the search condition includes more than one predicate, the predicates are linked with a logical AND (to narrow the search) or OR (to broaden it). The following example shows how you can use multiple conditions when searching an employee table to find the employee (or employees) with the specified first and last names:

```
WHERE lname = 'Smith' AND fname = 'Jean'
```

A single predicate follows this format:

```
search_expression operator search_value
```

In most instances, *search_expression* is the name of a column to search. Similarly, the most common form of *search_value* is a literal value to search for, which can be either a string of characters or a number.

The following two examples show literal values. The first searches for all the employees who are in the United Kingdom, and the second searches for all employees with a specific job level:

```
WHERE region = 'UK'
```

```
WHERE job_lvl = 100
```

Both *search_expression* and *search_value* can consist of any (or any combination) of the following:

- **Literal** A single text, numeric, date, or logical value. The following

example uses a literal to find all rows for employees in the United Kingdom:

```
WHERE region = 'UK'
```

- **Column reference** The name of a column in one of the tables being searched. The following example searches a products table for all rows in which the value of the production cost is lower than the shipping cost:

```
WHERE prod_cost < ship_cost
```

- **Function** A reference to a function that the database back end can resolve to calculate a value for the search. The function can be a function defined by the database server or a user-defined function that returns a scalar value. The following example searches for orders placed today (the GETDATE() function returns the current date):

```
WHERE order_date = GETDATE()
```

- **NULL** The following example searches an authors table for all authors who have a first name on file:

```
WHERE au_fname IS NOT NULL
```

- **Calculation** The result of a calculation that can involve literals, column references, or other expressions. The following example searches a products table to find all rows in which the retail sales price is more than twice the production cost:

```
WHERE sales_price > (prod_cost * 2)
```

- **Subquery** A result set generated by another query. The following example searches a products table to find all the products from Swedish suppliers. The subquery first searches the suppliers table to build a list of the suppliers located in that country/region. The second search then searches the products table, matching the product's supplier ID against the list created by the subquery:

```
WHERE supplier_id IN  
(SELECT supplier.supplier_id
```

```
FROM supplier
WHERE (supplier.country = 'Sweden'))
```

[Including or excluding columns](#)

You can choose which columns appear in a query result. When choosing which columns to include, there are several things to keep in mind:

- **You can include all of a table's columns** For example, you can include everything about each employee. The resulting SQL looks like this:

```
SELECT *
FROM employee
```

- **You can include exactly the columns you want** For example, you can list the name of all the employees. The resulting SQL looks like this:

```
SELECT fname, minit, lname
FROM employee
```

The list of columns you include might not even include a column from every table in the query. This does not mean that the table does not contribute to the query.

- **You can include all columns from all tables** For example, when you combine data from the sales and stores tables, you can include every column from either table in the result. The resulting SQL might look like this:

```
SELECT      *
FROM        sales INNER JOIN
stores ON   sales.stor_id = stores.stor_id
```

- **You can include derived columns** That is, you can include columns that are not part of any database table of the query. For example, you can create a result set containing the job description and the average job level for each job. The resulting SQL might look like this:

```

SELECT    job_desc, (max_lvl + min_lvl) / 2
FROM      jobs

```

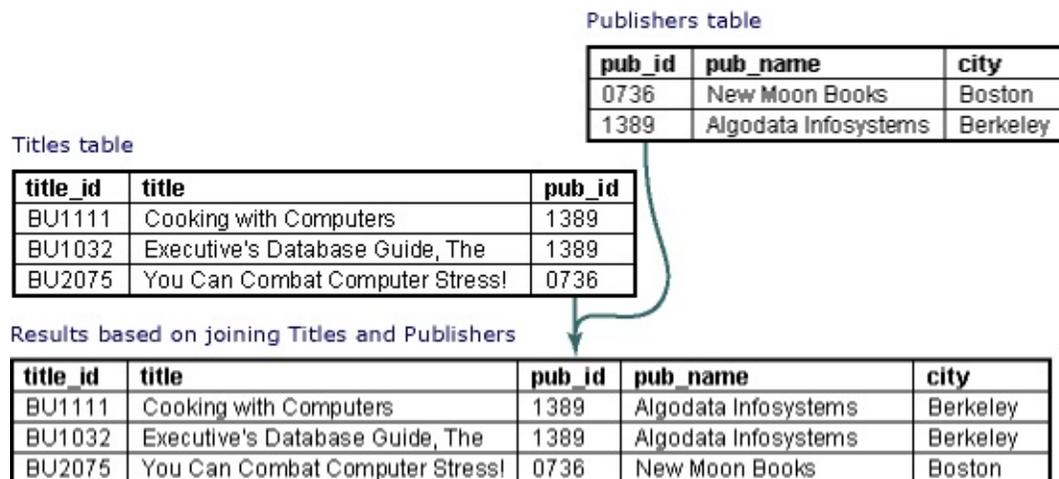
You can use SQL syntax to define the derived column (as in the preceding sample query) or you can employ a user-defined function that returns a scalar value.

[Combining tables](#)

A query result can include data from multiple tables. To combine data from tables, you use the Join operation from SQL.

[The basic join operation](#)

The Join operation matches rows of one table with rows of another table, based on values in those rows. For example, you can join the table `titles` with the table `publishers`. Each row in the result set will describe a title, including information about that title's publisher, as shown in the following illustration:



[Types of joins](#)

When you join tables, the type of join that you create affects the rows that appear in the result set. You can create the following types of joins:

- **Inner join** A join that displays only the rows that have a match in both joined tables. (This is the default type of join in the Query Designer.) For

example, you can join the `titles` and `publishers` tables to create a result set that shows the publisher name for each title. In an inner join, titles for which you do not have publisher information are not included in the result set, nor are publishers with no titles. The resulting SQL for such a join might look like this:

```
SELECT title, pub_name
FROM    titles INNER JOIN publishers
        ON titles.pub_id = publishers.pub_id
```

Note Columns containing NULL do not match any values when you are creating an inner join and are therefore excluded from the result set. Null values do not match other null values.

- **Outer join** A join that includes rows even if they do not have related rows in the joined table. You can create three variations of an outer join to specify the unmatched rows to be included:

- **Left outer join** All rows from the first-named table (the "left" table, which appears leftmost in the JOIN clause) are included. Unmatched rows in the right table do not appear. For example, the following SQL statement illustrates a left outer join between the `titles` and `publishers` tables to include all titles, even those you do not have publisher information for:

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
FROM titles LEFT OUTER JOIN publishers
        ON titles.pub_id
           = publishers.pub_id
```

- **Right outer join** All rows in the second-named table (the "right" table, which appears rightmost in the JOIN clause) are included. Unmatched rows in the left table are not included. For example, a right outer join between the `titles` and `publishers` tables will include all publishers, even those who have no titles in the `titles` table. The resulting SQL might look like this:

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
```

```
FROM titles RIGHT OUTER JOIN publishers
      ON titles.pub_id
      = publishers.pub_id
```

- **Full outer join** All rows in all joined tables are included, whether they are matched or not. For example, a full outer join between `titles` and `publishers` shows all titles and all publishers, even those that have no match in the other table.

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
FROM titles FULL OUTER JOIN publishers
      ON titles.pub_id
      = publishers.pub_id
```

- **Cross join** A join whose result set includes one row for each possible pairing of rows from the two tables. For example, `authors CROSS JOIN publishers` yields a result set with one row for each possible author/publisher combination. The resulting SQL might look like this:

```
SELECT *
FROM authors CROSS JOIN publishers
```

[Joining columns](#)

The JOIN operator matches rows by comparing values in one table with values in another. You decide which columns from each table should be matched. You have several choices:

- **Related Columns** Typically, you join tables by matching values in columns for which a foreign-key relationship exists. For example, you can join `discounts` to `stores` by matching the values of `stor_id` in the respective tables. The resulting SQL might look like this:

```
SELECT *
FROM discounts INNER JOIN stores
      ON stores.stor_id = discounts.stor_id
```

- **Unrelated Columns** You can also join tables by matching values in

columns for which no foreign-key relationship exists. For example, you can join publishers to authors by matching the values of state in the respective tables. Such a join yields a result set in which each row describes an author-publisher pair located in the same state.

```
SELECT au_lname,  
       au_fname,  
       pub_name,  
       authors.state  
FROM   authors INNER JOIN publishers  
       ON authors.state  
       = publishers.state
```

Note also that you use multiple columns to match rows from the joined tables. For example, to find the author-publisher pairs in which the author and publisher are located in the same city, you use a join operation matching the respective state columns and the respective city columns of the two tables. You need to match both city and state because it is possible that different states could have like-named cities (e.g., Springfield, Illinois and Springfield, Massachusetts).

[Join comparison operators](#)

The JOIN operator matches rows by comparing values in one table with values in another. You can decide exactly what constitutes a match. Your choices fall into two broad categories:

- **Match on Equality** Typically, you match rows when the respective column values are equal. For example, to create a result set in which each row contains a full description of each publisher (that is, with columns from the publishers table and the pub_info table), you use a join matching rows where the values of pub_id in the respective tables are equal. The resulting SQL might look like this:

```
SELECT *  
FROM   publishers INNER JOIN pub_info  
       ON publishers.pub_id  
       = pub_info.pub_id
```

- **Other** You can match rows using some test other than equality. For

example, to find the employees and the jobs for which they are underqualified, you can join employee with jobs, matching rows in which the job's minimum required level exceeds the employee's job level. The resulting SQL might look like this:

```
SELECT
  fname, minit, lname,
  job_desc, job_lvl, min_lvl
FROM employee INNER JOIN jobs
ON employee.job_lvl
  < jobs.min_lvl
```

[Joining tables](#)

When combining data from multiple tables, you must decide what tables to use. There are several noteworthy considerations:

- **Combining Three or More Tables** Each JOIN operation combines two tables. However, you can use multiple JOIN operations within one query to assemble data from any number of tables. Because the result of each JOIN operation is effectively a table, you can use that result as an operand in a subsequent join operation. For example, to create a result set in which each row contains a book title, an author, and the percentage of that book's royalties the author receives, you must combine data from three tables: authors, titles, and titleauthor. The resulting SQL might look like this:

```
SELECT
  title,
  au_fname,
  au_lname,
  royaltyper
FROM
  authors
  INNER JOIN
  titleauthor
  ON authors.au_id
  = titleauthor.au_id
  INNER JOIN
  titles
  ON titleauthor.title_id
  = titles.title_id
```

- **Using a Table merely to join others** You can include a table in a join even if you do not want to include any of that table's columns in a result set. For example, to establish a result set in which each row describes a title-store pair in which that store sells that title, you include columns from two tables: titles, and stores. But you must use a third table, sales, to determine which stores have sold which titles. The resulting SQL might look like this:

```
SELECT title, stor_name
FROM titles
      INNER JOIN
      sales
      ON titles.title_id = sales.title_id
      INNER JOIN
      stores
      ON
      sales.stor_id = stores.stor_id
```

Notice that the sales table contributes no columns to the result set.

- **Using a table twice in one query** You can use the same table two (or more) times within a single query.
- **Using something else in place of a table** In place of a table, you can use a query, a view, or a user-defined function that returns a table.

[Collapsing groups of rows](#)

You can create a query result in which each result row corresponds to an entire group of rows from the original data. When collapsing rows, there are several things to keep in mind:

- **You can eliminate duplicate rows** Some queries can create result sets in which multiple identical rows appear. For example, you can create a result set in which each row contains the city and state name of a city containing an author – but if a city contains several authors, there will be several identical rows. The resulting SQL might look like this:

```
SELECT city, state
FROM authors
```

The result set generated by the preceding query is not very useful. If a city

contains four authors, the result set will include four identical rows. Since the result set does not include any columns other than city and state, there is no way to distinguish the identical rows from each other. One way to avoid such duplicate rows is to include additional columns that can make the rows different. For example, if you include author name, each row will be different (provided no two like-named authors live within any one city). The resulting SQL might look like this:

```
SELECT city, state, fname, minit, lname
FROM authors
```

Of course, the preceding query eliminates the symptom, but does not really solve the problem. That is, the result set has no duplicates, but it is no longer a result set about cities. To eliminate duplicates in the original result set, and still have each row describe a city, you can create a query returning only distinct rows. The resulting SQL might look like this:

```
SELECT DISTINCT city, state
FROM authors
```

- **You can calculate on groups of rows** That is, you can summarize information in groups of rows. For example, you can create a result set in which each row contains the city and state name of a city containing an author, plus a count of the number of authors contained in that city. The resulting SQL might look like this:

```
SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
```

- **You can use selection criteria to include groups of rows** For example, you can create a result set in which each row contains the city and state name of a city containing several authors, plus a count of the number of authors contained in that city. The resulting SQL might look like this:

```
SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
```

HAVING COUNT(*) > 1

[Using a table twice in one query](#)

You can use the same table two (or more) times within a single query. There are several situations in which you do this.

- **Creating a self-join with a reflexive relationship** You can join a table to itself using a reflexive relationship—a relationship in which the referring foreign-key columns and the referred-to primary-key columns are in the same table. For example, suppose the employee table contains an additional column, employee.manager_emp_id, and that a foreign key exists from manager_emp_id to employee.emp_id. Within each row of the employee table, the manager_emp_id column indicates the employee's boss. More precisely, it indicates the employee's boss's emp_id.

By joining the table to itself using this reflexive relationship, you can establish a result set in which each row contains a boss's name and the name of one of that boss's employees. The resulting SQL might look like this:

```
SELECT
    boss.lname,
    boss.fname,
    employee.lname,
    employee.fname
FROM
    employee
    INNER JOIN
    employee boss
    ON employee.manager_emp_id
       = boss.emp_id
```

- **Creating a self-join without a reflexive relationship** You can join a table to itself without using a reflexive relationship. For example, you can establish a result set in which each row describes an employee and a potential mentor for that employee. (A potential mentor is an employee with a higher job level.) The resulting SQL might look like this:

```

SELECT
    employee.fname,
    employee.lname,
    mentor.fname,
    mentor.lname
FROM
    employee
    INNER JOIN
    employee mentor
    ON employee.job_lvl
    < mentor.job_lvl

```

Notice that the join uses a condition other than equality.

- **Using a table twice without a self-join** Even without a self-join, you can use the same table twice (or more) in a query. For example, you can establish a result set containing the other books by the author or authors of your favorite book. In this case, you use the titleauthors table twice— once to find the authors of your favorite book (*Is Anger the Enemy?*), and once to find the other books by those authors. The resulting SQL might look like this:

```

SELECT
    other_title.title
FROM
    titles favorite_title
    INNER JOIN
    titleauthor favorite_titleauthor
    ON favorite_title.title_id
    = favorite_titleauthor.title_id
    INNER JOIN
    authors
    ON favorite_titleauthor.au_id
    = authors.au_id
    INNER JOIN
    titleauthor other_titleauthor
    ON authors.au_id
    = other_titleauthor.au_id
    INNER JOIN
    titles other_title
    ON other_titleauthor.title_id
    = other_title.title_id
WHERE
    favorite_title.title
    = 'Is Anger the Enemy?'

```

```
AND
  favorite_title.title
  <> other_title.title
```

Note To distinguish between the multiple uses of any one table, the preceding query uses the following aliases: favorite_title, favorite_titleauthor, other_titleauthor, and other_title.

[Using views, user-defined functions, and subqueries in place of a table](#)

Whenever you write a query, you articulate what columns you want, what rows you want, and where the query processor should find the original data. Typically, this original data consists of a table or several tables joined together. But the original data can come from sources other than tables. In fact, it can come from views, user-defined functions that return a table, or subqueries.

[Using a view in place of a table](#)

You can select rows from a view. For example, suppose the database includes a view called “ExpensiveBooks,” in which each row describes a title whose price exceeds 19.99. The view definition might look like this:

```
SELECT *
FROM titles
WHERE price > 19.99
```

You can select the expensive psychology books merely by selecting the psychology books from the ExpensiveBooks view. The resulting SQL might look like this:

```
SELECT *
FROM ExpensiveBooks
WHERE type = 'psychology'
```

Similarly, a view can participate in a JOIN operation. For example, you can find the sales of expensive books merely by joining the sales table to the

ExpensiveBooks view. The resulting SQL might look like this:

```
SELECT *
FROM sales
  INNER JOIN
  ExpensiveBooks
  ON sales.title_id
  = ExpensiveBooks.title_id
```

[Using a user-defined function in place of a table](#)

In Microsoft SQL Server 2000, you can create a user-defined function that returns a table. Such functions are useful for performing complex or procedural logic.

For example, suppose the employee table contains an additional column, `employee.manager_emp_id`, and that a foreign key exists from `manager_emp_id` to `employee.emp_id`. Within each row of the employee table, the `manager_emp_id` column indicates the employee's boss. More precisely, it indicates the employee's boss's `emp_id`. You can create a user-defined function that returns a table containing one row for each employee working within a particular high-level manager's organizational hierarchy. You might call the function `fn_GetWholeTeam`, and design it to take an input variable—the `emp_id` of the manager whose team you want to retrieve.

You can write a query that uses the `fn_GetWholeTeam` function as a source of data. The resulting SQL might look like this:

```
SELECT *
FROM
  fn_GetWholeTeam ('VPA30890F')
```

“VPA30890F” is the `emp_id` of the manager whose organization you want to retrieve.

[Using a subquery in place of a table](#)

You can select rows from a subquery. For example, suppose you have already

written a query retrieving titles and identifiers of the coauthored books—the books with more than one author. The SQL might look like this:

```
SELECT
  titles.title_id, title, type
FROM
  titleauthor
  INNER JOIN
  titles
  ON titleauthor.title_id
  = titles.title_id
GROUP BY
  titles.title_id, title, type
HAVING COUNT(*) > 1
```

You can then write another query that builds on this result. For example, you can write a query that retrieves the coauthored psychology books. To write this new query, you can use the existing query as the source of the new query's data. The resulting SQL might look like this:

```
SELECT
  title
FROM
  (
    SELECT
      titles.title_id,
      title,
      type
    FROM
      titleauthor
      INNER JOIN
      titles
      ON titleauthor.title_id
      = titles.title_id
    GROUP BY
      titles.title_id,
      title,
      type
    HAVING COUNT(*) > 1
  )
  co_authored_books
WHERE type = 'psychology'
```

The emphasized text shows the existing query used as the source of the new query's data. Note that the new query uses an alias ("co_authored_books") for the existing query.

Similarly, a query can participate in a JOIN operation. For example, you can find the sales of expensive coauthored books merely by joining the ExpensiveBooks view to the query retrieving the coauthored books. The resulting SQL might look like this:

```
SELECT
  ExpensiveBooks.title
FROM
  ExpensiveBooks
  INNER JOIN
  (
    SELECT
      titles.title_id,
      title,
      type
    FROM
      titleauthor
      INNER JOIN
      titles
    ON titleauthor.title_id
    = titles.title_id
    GROUP BY
      titles.title_id,
      title,
      type
    HAVING COUNT(*) > 1
  )
```

[Sorting rows](#)

You can order the rows in a query result. That is, you can name a particular column or set of columns whose values determine the order of rows in the result set. There are several ways in which you can use ordering:

- **You can arrange rows in ascending or descending order** By default, SQL uses order-by columns to arrange rows in ascending order. For example, to arrange the book titles by ascending price, simply sort the rows by the price column. The resulting SQL might look like this:

```
SELECT *
FROM titles
ORDER BY price
```

On the other hand, if you want to arrange the titles with the more expensive books first, you can explicitly specify a highest-first ordering. That is, you indicate that the result rows should be arranged by descending values of the price column. The resulting SQL might look like this:

```
SELECT *
FROM titles
ORDER BY price DESC
```

- **You can sort by multiple columns** For example, you can create a result set with one row for each author, ordering first by state and then by city. The resulting SQL might look like this:

```
SELECT *
FROM authors
ORDER BY state, city
```

- **You can sort by columns not appearing in the result set** For example, you can create a result set with the most expensive titles first, even though the prices do not appear. The resulting SQL might look like this:

```
SELECT title_id, title
FROM titles
ORDER BY price DESC
```

- **You can sort by derived columns** For example, you can create a result set in which each row contains a book title— with the books that pay the highest royalty per copy appearing first. The resulting SQL might look like this:

```
SELECT title, price * royalty / 100 as royalty_per_unit
FROM titles
ORDER BY royalty_per_unit DESC
```

(The formula for calculating the royalty that each book earns per copy is emphasized.)

To calculate a derived column, you can use SQL syntax, as in the preceding example, or you can use a user-defined function that returns a scalar value.

- **You can sort grouped rows** For example, you can create a result set in which each row describes a city, plus the number of authors in that city — with the cities containing many authors appearing first. The resulting SQL might look like this:

```
SELECT city, state, COUNT(*)
FROM authors
GROUP BY city, state
ORDER BY COUNT(*) DESC, state
```

Notice that the query uses state as a secondary sort column. Thus, if two states have the same number of authors, those states will appear in alphabetical order.

- **You can sort using international data** That is, you can sort a column using collating conventions that differ from the default conventions for that column. For example, you can write a query that retrieves all the book titles by the Icelandic novelist Halldor Laxness. To display the titles in alphabetical order, you use an Icelandic collating sequence for the title column. The resulting SQL might look like this:

```
SELECT title
FROM
    authors
    INNER JOIN
        titleauthor
        ON authors.au_id
        = titleauthor.au_id
    INNER JOIN
        titles
        ON titleauthor.title_id
        = titles.title_id
WHERE
    au_fname = 'Halldor' AND
```

```
    au_lname = 'Laxness'  
ORDER BY  
    title COLLATE SQL_Icelandic_Pref_CP1_CI_AS
```

[Show All](#)

Add columns to a query (ADP)

[Background information](#)

To use a column in a query, you must add it to the query. You might add a column to display it in query output, to use it for sorting, to search the contents of the column, or to summarize its contents.

If you are creating a Select query and add a column in the Diagram or Grid panes, the column you add becomes part of the query output. You can remove the column from the output and still use it for sorting, searching, and so on. For example, to find all employees in the accounting department, you might search the department column but not display it in the output.

[Tip](#)

Wherever you use a column in a query, you can also use an expression that can consist of any combination of columns, literals, operators, and functions.

You can add columns individually or as a group. Your choices are:

- An individual column from one table, view, or function, to use for sorting, searching, or summarizing.
- All columns from one table, view, or function. For tables, this is the equivalent of specifying "*tablename.**" in the SQL statement.
- All columns from all tables, views, or functions in the query, which can be useful if you are working with joins. This option is the equivalent of specifying "*" alone in the SQL statement when more than one table, view, or function is used in the query.

When you add all columns, the Query Designer does not add all individual columns to the query; it instead uses the asterisk ("*"). If you need to work with a specific column, you must add it separately.

Note When you choose "*", all currently defined columns for the tables you are using are included. If a table definition changes, the list of columns returned by "*" changes as well. It is recommended that, if practical, you

specify the columns you want to work with instead of using "*".

[Add an individual column](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Diagram** pane, select the check box next to the data column that you want to include.

-or-

In the **Grid** pane, move to the first blank grid row where you want to add the column, click the field in the **Column** column, and select a column name from the list.

Note To add a data row at a specific location in the **Grid** pane, select the grid row where you want to add the new column and press **INS**. A new column is added above that row.

[Add all columns for one table, view, or function](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Diagram** pane, select the check box next to **(All Columns)**.

-or-

Specify *objectname.** in the SQL SELECT statement in the SQL pane, substituting the name of your table, view, or function for *objectname*.

[Add all columns for all tables, views, or functions](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Make sure no join lines in the **Diagram** pane are selected.
3. Right-click in the query window and choose **Properties** from the shortcut menu.
4. Select **Output all columns**.

- or -

Specify * in the output list of the SQL SELECT statement in the SQL pane.

[Show All](#)

About Query Designer considerations for SQL Server databases (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The following guidelines provide information about Microsoft SQL Server-specific features that you can use.

[SQL Server versions](#)

The Query Designer is designed to support versions of Microsoft SQL Server 2000, but is compatible with earlier versions as well. If you attempt to use a SQL Server 2000 feature when connected to version 7.0 or 6.5 of SQL Server, the server will report an error.

[SQL syntax](#)

When the Query Designer builds a statement in the SQL pane, it will use syntax specific to SQL Server whenever possible. For example, database objects such as tables and views are qualified using SQL Server owner names.

You can also type SQL Server-specific syntax in the SQL pane. In some cases when you verify a query, the Query Designer converts server-specific syntax to ANSI standard syntax. However, the changed query will always return the same results.

[Identifying database objects](#)

When you enter the names of database objects (tables, views, and columns) in the SQL pane, you must provide sufficient information for SQL Server to identify the object you want. Database objects are identified with unique names that consist of up to three parts (for tables and views) or four parts (for columns):

`database.owner.table`

database.owner.table.column

Note In Microsoft SQL Server 7.0 or higher, you can join tables from different databases on the same server. In that case, database objects can have four part names.

In general, you need to provide only enough qualifiers to uniquely identify the object you want to work with. For example, if you are working with a column called price in the titles table in the current database, you can simply reference the column by name, as in this SQL statement:

```
SELECT price
FROM titles
```

However, if you are working with two tables, such as orders and products, and each has a column called price, you must qualify references to the column with the appropriate table name, as in this example:

```
SELECT products.prod_id, orders.price
FROM orders INNER JOIN products ON
orders.prod_id = products.prod_id
```

When you use the Diagram pane and Grid pane to work with tables in the current database, the Query Designer automatically adds owner and table qualifiers for you. If you are not the owner of a table that you are working with, the owner's name will appear in the table names. For example, if you work in the pubs database, the owner name dbo will appear in front of table names. If you are working with multiple tables, the Query Designer adds table name qualifiers to column names.

[Using quotation marks](#)

The standard delimiters for literal strings in SQL are single quotation marks ('). By default, SQL Server reserves double quotation marks (") as delimiters for database objects.

To avoid ambiguity, the Query Designer always sets Quoted Identifiers on, so that double quotation marks are always interpreted as database object delimiters. If you have previously turned Quoted Identifiers off, the Query Designer

overrides your setting.

Therefore, in the Query Designer, always use single quotation marks to enclose string literals. Use double quotation marks only as needed for database objects delimiters.

[Case sensitivity](#)

Text information in a SQL Server database can be stored in uppercase letters, lowercase letters, or a combination of both. For example, a last name can appear as "SMITH," "Smith," or "smith."

Depending on how SQL Server was installed, databases can be case-sensitive or case-insensitive. If a database is case-sensitive, when you search for text data, you must construct your search conditions using the exact combination of uppercase and lowercase letters. For example, if you are looking for a name such as "Smith," you cannot use the search conditions "=smith" or "=SMITH."

In addition, if the server was installed as case-sensitive, you must provide database, owner, table, and column names using the correct combination of uppercase and lowercase characters. If the case of the name you provide does not match exactly, SQL Server returns an error reporting an "invalid object name."

When you create queries using the Diagram and Grid panes, the Query Designer will always accurately reflect the case-sensitivity of your server. However, if you enter queries in the SQL pane, you must be careful to match names to the way they will be interpreted by the server.

If the server was installed with a case-insensitive option, you can enter database object identifiers and search conditions using any combination of uppercase and lowercase characters.

[Tip](#)

To determine the case sensitivity of a server, execute the system stored procedure `sp_server_info`, and then examine the contents of row 18. If the server has been installed with the case-insensitive setting, the option for `sort_order` will be set to `nocase`.

[Entering keywords in the Grid and SQL panes](#)

The Query Designer supports the use of certain SQL Server constants, variables, and reserved column names in the Grid or SQL panes. Generally, you can enter these values by typing them in, but the Grid pane will not display them in drop-down lists. Examples of supported names include:

- **IDENTITYCOL** If you enter this name in the Grid or SQL pane, the SQL Server will recognize it as a reference to an auto-incrementing column.
- **Predefined global values** You can enter values such as @@CONNECTIONS and @@CURSOR_ROW into the Grid and SQL panes. For a complete list of global values, see the SQL Server documentation.
- **Constants (niladic functions)** You can enter constant values such as CURRENT_TIMESTAMP and CURRENT_USER in either pane. For a complete list of constants (niladic functions), see the SQL Server documentation.
- **NULL** If you enter NULL in the Grid or SQL panes, it is treated as a literal value, not a constant.
- **ROWGUIDCOL** If you enter this name in the Grid or SQL pane, the SQL Server will recognize the column as a row global unique identifier column. For more information on row global unique identifier columns, see the SQL Server documentation.

[Entering currency values](#)

In the Grid pane, to specify that you want data interpreted as money, precede the value with \$ or \$- (for negative values). Do not include a comma or other delimiter to indicate thousands. Formatting values this way alerts the Query Designer that you are entering values to be treated as or compared to data in money or smallmoney type columns. Values are rounded to the nearest hundredth of a unit.

You can use \$ no matter what currency you are working with. When a query displays values from money columns in datasheet view, it does not include the \$ prefix. Depending on the setting in the **Windows Regional Settings** dialog box, currency data might or might not include a comma or other delimiter for thousands.

[Using the GUID data type](#)

In Microsoft SQL Server 7.0 or higher, you can include references to the GUID data type, which is used to store globally unique identifiers. In Update and Append queries you can call the NEWID() function to generate a new GUID to be stored in the database.

When you are creating a Select query, the only operations allowed with a GUID type column are comparisons based on equality (= and <>).

[Entering blanks](#)

If you are working with Microsoft SQL Server 7.0 or higher, you can specify a zero-length string in an Update, Append Values, or Make-Table query by entering two single quotation marks, as in the following Update query example:

```
UPDATE employee
SET minit = ''
WHERE emp_id = 'CFS88322F'
```

In versions of SQL Server 6.5 or earlier, two single quotation marks are treated as a single space. For example, you can use quotation marks in the following expression: 'abc' + ' ' + 'def'. The resulting value would be 'abc def'.

[Including optimizer hints](#)

If you are entering a query directly in the SQL pane, you can add optimizer hints to specify the use of specific indexes, locking methods, and so on. However, when reformatting the contents of the SQL pane, the Query Designer might not maintain these comments. Optimizer comments are not represented graphically.

For more details about optimizer hints, refer to the Microsoft SQL Server documentation.

[ANSI to OEM character conversion](#)

Data containing extended characters—that is, characters outside the ASCII range 32 (space) to 126 (~), including international characters such as "ä," "ç," "é," "ñ," and "ß"—can require special handling when you are working with SQL Server.

The representation of extended characters in a result set depends on the code page in use. A code page is a character set that a computer uses to interpret and display data properly. Code pages usually correspond to different platforms and languages and are used in international applications. For example, the ASCII value 174 might appear as the symbol "®" in one code page but as a chevron character in another code page.

In general, code pages are divided into ANSI code pages and OEM code pages. ANSI code pages, in which high-numbered ASCII values represent international characters, are used in Windows. OEM code pages, in which high-numbered ASCII values represent line-drawing and punctuation characters, were designed for MS-DOS.

When data is entered into a SQL Server database, SQL Server settings on the local (client) computer specify whether the data is stored in ANSI or OEM format. The option is specified using the **Automatic ANSI to OEM conversion** option on **the DB Library Options** tab in the **SQL Server Client Configuration** dialog box. This dialog box is available by clicking the Microsoft SQL Server 7.0 or later **Client Network Utility** from Programs on the **Start** menu. (For more details about Automatic ANSI to OEM conversion, refer to the SQL Server documentation.)

By default, this option is selected for the SQL Server Client, a choice which causes the data to be converted from high-numbered ASCII characters to OEM characters. For example, if the OEM conversion option is set and you enter the name "Günther" in a column and then save the row, the character "ü" will be converted to another character before the row is stored in the database.

The results of queries that you create in the Query Designer are affected by the format in which extended-character data is stored in combination with the setting of the OEM conversion option in the **SQL Server Client Configuration** dialog box. Depending on these variables:

- You might not be able to search for data that includes high-order ASCII characters.
- Your query results might appear with incorrect characters substituted for high-order ASCII characters.

In general, if data is stored in OEM format, you should set the OEM conversion

option so the data will display properly and so you can search it. If data is stored in ANSI format (that is, it was not converted to OEM format) but you have set the OEM conversion option, the data will not display properly and you will not be able to search for it.

To determine whether data was stored in OEM format, you can use a query to display the contents of the table or tables you are working with. If extended characters appear incorrectly, the OEM conversion setting is probably wrong. Close the query and the Access project, change the setting in the **SQL Server Client Configuration** dialog box, and then open the Access project and query again.

[Unsupported and partially supported query types](#)

Some types of legal SQL Server queries cannot be represented graphically in the Query Designer. You can still enter them in the SQL pane, and they will execute correctly. However, the Query Designer will display the **Query Definitions Differ** dialog box and report an error when you execute your query or change panes.

Several types of SQL Server queries are not supported graphically, including:

- Queries using INTERSECT.
- Queries using UNION [ALL].
- Queries using CASE.
- Any data definition (DDL) query, including CREATE TABLE, ALTER TABLE, CREATE PROCEDURE, ALTER PROCEDURE, and so on. CREATE VIEW and ALTER VIEW queries are not supported graphically, but you can use the Query Designer to create and edit views.
- Update and Delete queries that include an extra FROM clause (FROM *table* FROM *table*) that specifies the list of rows to update or delete.
- Queries using the FOR BROWSE clause.
- Queries that include UPDATE as a search condition.
- Queries including CURRENT OF.

Add a comment to a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can document a view, user-defined function, or single statement stored procedure by adding a comment to it. You can enter a comment in a property page.

1. In the Database window, click **Queries**  under **Objects**, click the view, user-defined function, or single statement stored procedure you want to open, and then click **Design** on the database window toolbar.
2. Right-click in the Query Designer, then select **Properties** from the shortcut menu.
3. Enter the comment in the **SQL Comment** field.

Notes

- You should avoid entering the two-character strings "/"* and "*/" within your comments.
- If you change the type of query (for example, from a SELECT query to an UPDATE query), the query's comment is deleted.

[Show All](#)

Add tables, views, and functions to a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.



[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL**  to show the **SQL** pane.
2. In the SQL pane, type a right and left parentheses () after the word FROM.
3. Open the Query designer for the existing query . (You now have two Query Designers open.)
4. Display the SQL pane for the inner query— the existing query you are including in the new, outer query.
5. Select all the text in the SQL pane, and copy it to the Clipboard.
6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.

Create column aliases in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can create aliases for column names to make it easier to work with column names, calculations, and summary values. For example, you can create a column alias to:

- Create a column name, such as "Total Amount," for an expression such as (quantity * unit_price) or for an aggregate function.
- Create a shortened form of a column name, such as "d_id" for "discounts.stor_id."

After you have defined a column alias, you can use the alias in a Select query to specify query output.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the **Grid** pane, locate the row containing the data column for which you want to create an alias, and if necessary, mark it for output. If the data column is not already in the grid, add it.
3. In the **Alias** column for that row, enter the alias. The alias must follow all naming conventions for SQL. If the alias name you enter contains spaces, the Query Designer automatically puts delimiters around it.

[Show All](#)

Create a subquery (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can use the results of one query as the input for another. Typically, you use the results of a subquery as a search condition that uses the IN() function or EXISTS operator. However, you can also use a subquery in the FROM clause.

You can create a subquery by entering it in either the Grid pane or SQL pane.

[Define an EXISTS subquery in the Grid pane](#)

1. In the Database window, click **Queries**  under **Objects**, click the primary query you want to open, and then click **Design** on the database window toolbar.
2. In the **Column** column for the first empty row in the **Grid** pane, enter EXISTS followed by the subquery in parentheses.
3. In the **Criteria** column for the row containing the subquery, enter TRUE, FALSE, =TRUE, or =FALSE. Entering FALSE or =FALSE results in a NOT EXISTS query.

Note To create a NOT EXISTS query, create an EXISTS query as listed in the above steps, and set the Criteria column to FALSE. If you enter NOT EXISTS in the Grid pane, the Query Designer will display an error.

[Define a subquery in the SQL pane](#)

1. In the Database window, click **Queries**  under **Objects**, click the primary query you want to open, and then click **Design** on the database window toolbar.
2. If necessary, click **SQL**  to show the **SQL** pane.
3. In the **SQL** pane, select the SQL statement, and then use **Copy** to move the query to the Clipboard.
4. Start the new query, and then use **Paste** to move the first query into the new query's WHERE or FROM clause.

For example, imagine you have two tables, `products` and `suppliers`, and you want to create a query showing all products for suppliers in Sweden. Create the first query on the `suppliers` table to find all Swedish suppliers:

```
SELECT supplier_id
FROM supplier
WHERE (country = 'Sweden')
```

Use the Copy command to move this query to the Clipboard. Create the second query using the `products` table, listing the information you need about products:

```
SELECT product_id, supplier_id, product_name
FROM products
```

In the SQL pane, add a WHERE clause to the second query, then paste the first query from the Clipboard. Place parentheses around the first query, so that the end result looks like this:

```
SELECT product_id, supplier_id, product_name
FROM products
WHERE supplier_id IN
  (SELECT supplier_id
   FROM supplier
   WHERE (country = 'Sweden'))
```

Note When you add a subquery to the WHERE clause, the subquery appears in the **Criteria** column of the Grid pane. You can edit it further in either the Grid pane or SQL pane. However, the tables, views, functions, columns, and expressions referenced in the subquery are not displayed in the Diagram or Grid pane.

Verify a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

To avoid problems, you can check the query you have built to ensure its syntax is correct. This option is especially useful when you enter statements in the SQL pane.

Note A statement can be valid, and therefore be verified successfully, even if it cannot be represented in the Diagram and Grid panes.

Note SQL Verification can detect some, but not all SQL errors. If a query contains an error not detected during SQL verification, the database will detect the error when you run the query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Click **Verify SQL Syntax** .

Create table aliases in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Aliases can make it easier to work with table view, or function names. Using aliases is helpful when:

- You want to make the statement in the SQL pane shorter and easier to read.
- You refer to the table, view, or function name often in your query— such as in qualifying column names— and want to be sure you stay within a specific character-length limit for your query.
- You are working with multiple instances of the same table, view, or function (such as in a self-join) and need a way to refer to one instance or the other.

For example, you can create an alias "e" for a table name `employee_information`, and then refer to the table as "e" throughout the rest of the query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram Pane, right-click the table, view, or function for which you want to create an alias, then select **Properties** from the shortcut menu.
3. In the **Properties** dialog box, enter the alias in the **Alias** box.

When you create a table alias, the Query Designer substitutes the alias for the corresponding table name in the Table column of the Grid pane.

Note The SQL standard specifies that when you create an alias for a table name, you must use the alias to refer to the table in the rest of the SQL statement (that is, you cannot use the original table name).

Exclude duplicate rows in a query (ADP)

If you want to see only unique values in a result set, you can specify that you want to exclude duplicates from the result set.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Right-click the background of the Diagram pane, then choose **Properties** from the shortcut menu.
3. Select **DISTINCT values**.

The Query Designer inserts the keyword DISTINCT in front of the list of display columns in the SQL statement.

Note If you use the DISTINCT keyword in Microsoft SQL Server, you cannot modify the data in datasheet view.

Remove tables, views, or functions from a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can remove a table, view, or function from a query.

Note Removing a table, view, or function does not delete anything from the database; it only removes it from the current query.

- In the Diagram pane, select the table, view, or function, and then press **DELETE**, or right-click the object and then choose **Remove**. You can select and remove multiple objects at one time.

–or–

- Remove all references to the object in the SQL pane.

When you remove a table, view, or function, the Query Designer automatically removes joins that involve that table, view, or function and removes references to the object's columns in the SQL and Grid panes. However, if the query contains complex expressions involving the object, the object is not automatically removed until all references to it are removed.

[Show All](#)

Reorder output columns in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

The order in which you add data columns to a Select query determines the order in which they appear in the results. The first column you add to the query appears leftmost in the results, the second column next, and so on.

If you are creating Update or Append queries, the order in which you add columns affects the order in which data is processed.

To control where a data column appears in the result set, or in what order it is used, you can reorder the columns.

[Reorder columns for output in the Grid pane](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Grid pane, select the row containing the column by clicking the row selector button to the left of the row (.
3. Point to the row selector button and drag the row to a new location.

[Reorder columns for output in the SQL pane](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. If necessary, click **SQL**  to show the **SQL** pane.
3. Edit the order of the column names in the SQL pane.

[Tip](#)

You can add a data row at a specific location in the Grid pane by inserting a blank row into the Grid pane, and then specifying the data column to insert.

[Show All](#)

Include user-defined functions in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Include an inline or table user-defined function in a query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Design the query to which you want to add the function. In the Query Designer, be sure the Diagram pane is visible.
3. From the list of functions, drag the function onto the Diagram pane.

[Include a scalar function in a query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Design the query to which you want to add the function. In the Query Designer, be sure the Grid pane is visible.
3. In the first blank row of the Grid pane, enter the name of the function in the **Column** column.

[Show All](#)

Run a parameter query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

1. Do one of the following:
 - In the Database window, click **Queries**  under **Objects**, click the [parameter query](#) you want to run, and then click **Open** on the database window toolbar.
 - In the Query Designer, Click **Run**  to run the parameter query.

Access shows the **Enter Parameter Value** dialog box for each parameter in the query.

2. Do one of the following for each parameter value you are prompted for:
 - To enter a parameter value, type the value.
 - To enter the [default value](#), select <DEFAULT>.
 - To enter a [Null](#) value, select <NULL>.
 - To enter a [zero-length](#) or blank string, leave the box blank.

Note You cannot run or create a [view](#) as a parameter query.

[Show All](#)

About different query types (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

A query is a specific request for the retrieval, creation, modification, or deletion of data in a database. There are two basic types of queries:

- A Select query is any query that retrieves data for display by using a SQL SELECT statement.
- An action query is a stored procedure that inserts, modifies, or deletes data, by using the SQL INSERT, UPDATE, and DELETE statements.

 [Views](#)

 [Using Views](#)

Storage Views are stored as part of a database design. As you design a database, you can include views in the design for the following reasons:

- **Some subsets of data are of interest to many users.** Because each view is stored in the database, it establishes a particular subset of data that can be used by any database user.
- **Views can conceal base tables.** You can disallow all user access to database tables, requiring users to manipulate data through views only. Such an approach can help protect users and application programs from certain database modifications. For example, you can create a view called “Current Month Sales.” On the first of each month, you can modify the view definition accordingly. Without such a view, users would need to rewrite their queries every month to select Sales rows from the appropriate month.

Sorting results You can sort a view, but only if the view includes the TOP clause.

Query plan generation A query plan is an internal strategy by which a

database server tries to create result sets quickly. A database server can establish a query plan for a view as soon as the view is saved.

[Using Indexed Views and Schema Binding](#)

[Indexed views and schema binding](#)

Indexed views are views whose results are persisted in the database and indexed for fast access. Indexed views are supported when your [Microsoft Access project](#) is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

As with any other views, indexed views depend on base tables for their data. Such dependency means that if you change a base table contributing to an indexed view, the indexed view might become invalid. For example, renaming a column that contributes to a view invalidates the view. To prevent such problems, SQL Server supports creating views with “schema binding.” Schema binding prohibits any table or column modification that would invalidate the view. Any indexed view you create with the Query Designer automatically gets schema binding, because SQL Server requires that indexed views have schema binding. Schema binding does not mean you cannot modify the view; it means you cannot modify the underlying tables or views in ways that would change the view’s result set.

[What happens when you modify a base table or column](#)

Using the Table Designer or Database Designer, you might attempt to modify a base table or column that contributes to a view defined with schema binding. If your attempted modification could invalidate the view, the Designer warns you and asks you whether you want to proceed. If you choose to proceed, these things happen:

- Your modifications to the base table occur.
- All views depending on the base table views are changed so that “schema

binding” is removed. Thus, your subsequent changes to the base table will proceed without warning.

- If the dependent views were indexed, the indexes are deleted.

The Table Designer and Database Designer warn you before modifying a base table only if that base table contributes to a view with schema binding and if your modification does one or more of the following:

- Deletes the base table
- Renames the base table
- Recreates the base table
- Removes a column from the base table and the view includes that column
- Renames a column from the base table and the view includes that column

[Stored Procedures](#)

Stored procedures can make managing your database and displaying information about that database and its users much easier. Stored procedures are a precompiled collection of SQL statements and optional control-of-flow statements stored under a name and processed as a unit. Stored procedures are stored within a database; can be executed with one call from an application; and allow user-declared variables, conditional execution, and other powerful programming features.

Stored procedures can contain program flow, logic, and queries against the database. They can accept parameters, output parameters, return single or multiple result sets, and return values.

You can use stored procedures for any purpose for which you would use SQL statements, with these advantages:

- You can execute a series of SQL statements in a single stored procedure.
- You can reference other stored procedures from within your stored procedure, which can simplify a series of complex statements.
- The stored procedure is compiled on the server when it is created, so it executes faster than individual SQL statements.

A single statement stored procedure contains only one SQL SELECT statement. You can graphically create or edit it in the Query Designer, and you can define

extended properties (except column properties).

A multistatement stored procedure contains more than one SQL statement. You can create or edit it in the SQL Text Editor, and you can even graphically edit an SQL SELECT statement in a multistatement stored procedure by using the Query Builder. However, you cannot define extended properties.

[User-Defined Functions](#)

When your Microsoft Access project is connected to a Microsoft SQL Server 2000 database, you can create and use user-defined functions. The following example shows the ListCust user-defined function that accepts a named parameter, returns a table, and is used in the FROM clause of a SELECT statement.

```
SELECT * FROM ListCust(@[Enter a Name])
```

[Overview of user-defined functions](#)

A user-defined function is a query, like a view or stored procedure, that you can:

- View in the database window.
- Create in the Query Designer or SQL Text Editor.
- Use to retrieve data and browse the results in a datasheet
- Use as the record source of a form, report, or combo box.

Note You cannot use a user-defined function as the record source of a data access page.

- Programmatically manipulate using the Access object model.

User-defined functions combine the best features of views and stored procedures into a single query that you can nest, pass parameters to, sort, and return values. In many cases, user-defined functions are a better alternative to views and stored procedures because you can return a single table of data or scalar value, hide the business logic and details of how that data or scalar value was created, and simplify the complexity of your SQL statement syntax.

For more information about and examples of user-defined functions, see the SQL Server documentation.

[The three types of user-defined functions](#)

Based on what kind of value it returns, each user-defined function falls into one of three categories:

in-line user-defined function Contains a single SELECT statement and returns an updateable table of data. You can use this function in the FROM clause of a query. You can graphically create and edit an inline user-defined function by using the Query Designer. You can also define extended properties.

table user-defined function Contains one or more SELECT statements and returns a non-updateable table of data. You can use this function in the FROM clause of a query. You can create and edit a table user-defined function by using the SQL Text Editor, and you can even graphically edit an SQL SELECT statement in a table user-defined function by using the Query Builder. However, you cannot define extended properties.

scalar user-defined function Contains one or more SELECT statements and returns a scalar value, such as the **int**, **decimal**, **varchar**, **sql_variant** or **table** data types. You can use a scalar function in a query anywhere you would use a column name. You can create and edit a scalar user-defined function by using the SQL Text Editor, and you can even graphically edit an SQL SELECT statement in a scalar user-defined function by using the Query Builder. However, you cannot define extended properties.

[Parameter Queries](#)

[What is a parameter query?](#)

If you want to repeatedly run the same query, but with different values at different times, you can use a parameter query. A parameter is a placeholder for a value that you type when the query runs. For example, if you want to find all the customers in a Customers table from the same country/region, but prompt for a different country/region each time, you can type @Enter_Country in the **Criteria** cell of the Design grid in Query Designer and generate the following SQL SELECT statement:

```
SELECT * FROM Customers
WHERE Country = @Enter_Country
```

The at sign (@) character in front of the parameter, Enter_Country, tells Access to show the **Enter Parameter** dialog box so you can type in a country/region name, such as "Mexico" or "Denmark", and use that as the value for the WHERE clause.

[Using parameters](#)

You can use parameters as placeholders for either text or numeric literal values. Usually, parameters are used as placeholders in search conditions for individual rows or for groups in the WHERE or HAVING clauses of an SQL statement.

You can even use a combination of parameters to specify a range of dates. For example, you can create two parameters - @Enter_Beginning_Date and @Enter_Ending_Date - in the WHERE clause of a query, and then specify a range of hire dates when you run the query as the following example shows.

```
SELECT * FROM ORDERS
WHERE ShippedDate
BETWEEN @Enter_Beginning_Date AND @Enter_Ending_Date
```

You can also use parameters as placeholders in expressions. For example, you might want to calculate discounted prices by supplying a different discount value each time you run a query. To do so, you could enter the expression UnitPrice * @Enter_Discount in the **Column** cell of the Design grid and generate the following SQL SELECT statement as the following expression shows:

```
SELECT ProductName, UnitPrice,
(UnitPrice * @Enter_Discount) AS DiscountPrice
FROM Products
```

[Comparing named and unnamed parameters](#)

You can specify two types of parameters, named and unnamed.

Named parameters

Named parameters are easier to read and use, and particularly useful if you have multiple parameters in a query and want to make clear what to enter in each parameter. For example, to prompt and search for an author's first and last names in an authors table, create the following SQL SELECT statement:

```
SELECT au_id  
FROM authors  
WHERE au_fname = @Enter_First_Name AND au_lname = @Enter_Last_Name
```

When you run the parameter query, Access prompts once for each parameter using the parameter name in the **Enter Parameter** dialog box so a user is clear which name to enter.

Unnamed parameters

You can also enter the question mark (?) character as an unnamed parameter.

For example, if you want to retrieve all authors from one state in a stored procedure, inline user-defined function, or SQL statement in the record source of a form or report, you can type the question mark (?) character in the **Criteria** cell of the Design grid in Query Designer and generate the following SQL SELECT statement:

```
SELECT au_lname, au_fname  
FROM state  
WHERE state = @Param1
```

In this case, Access will automatically give the unnamed parameter the generated name, @Param1.

Note Access does not provide a generated name in the case of an SQL Statement in a form or report's record source.



[When you can and can't use named parameters](#)

You can use named parameters under the following circumstances:

- When your Microsoft Access project is connected to Microsoft SQL Server database version 2000.
- In the Query Designer of a stored procedure or in-line user-defined function.
- In the SQL Text Editor for a stored procedure, scalar user-defined function, or table user-defined function.

You cannot use named parameters in the SQL statement of a record source that you create in the Query Builder.

You cannot use named or unnamed parameters in a [view](#).



[Action queries](#)

You can create the following types of action queries in the Diagram and Grid panes (the graphical panes) of the Query Designer:

- **Append Values query** Creates a new row and inserts literal values into specified columns. This type of query creates an SQL INSERT INTO... VALUES statement.
- **Append query** Creates new rows by copying existing rows from one table into another, or into the same table as new rows. This type of query creates an SQL INSERT...SELECT statement.
- **Update query** Changes the values of individual columns in one or more existing rows in a table. This type of query creates an SQL UPDATE statement.
- **Delete query** Removes one or more rows from a table. This type of query creates an SQL DELETE statement.

Note A Delete query removes entire rows from the table. If you want to delete values from individual data columns, use an Update query.

- **Make-Table query** Creates a new table and creates rows in it by copying the results of a query into it. This type of query creates an SQL SELECT... INTO statement.

[SQL SELECT Statements](#)

You can use a SQL statement as the record source of a form or report or as the row source of a list box or combo box on a form. You might do this to keep the query from being displayed in the database window. You can create the SQL SELECT statement by using the **Build**  button to the right of the field and graphically design the SQL SELECT statement in the Query Builder, or by editing your own SQL SELECT statement in any text editor and copying and pasting it into the **RecordSource** property field.

[Show All](#)

Create a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Create a view](#)

 [Create a non-indexed view](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design View**.
3. Design the view in the **Diagram** and **Grid** panes of the Query Designer.
4. Click **Save**  on the toolbar.
5. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules.

Microsoft Access adds the view to **Queries**  under **Objects** in the Database window.

 [Create an indexed view](#)

An indexed view is a view whose result set is stored in the database for fast access. Indexed views are supported when your [Microsoft Access project](#) is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add tables to the Diagram pane. Because you intend this to be an indexed view, be sure to add only tables that you own.

3. Select which columns you want included in the view. Do not use the asterisk (*); you must explicitly select each column that you want to appear in the indexed view.
4. Right-click the background of the Diagram pane, then choose **Manage Indexes** from the shortcut menu. The **Indexes** dialog box appears.

Note There are many situations in which the **Manage Indexes** command is unavailable, because there are many restrictions on which views can be indexed. For example, you cannot index a view unless you are the owner of each table contributing to it. For a complete description of the restrictions on creating indexed views, see the Microsoft SQL Server documentation.

5. Within the dialog box, click **New**.
6. Supply the information for the index definition— index name, index columns and their order, index file group, and the other index settings. For more information on each dialog box option, click **Help**.

[Create a stored procedure](#)

[Create a single statement stored procedure in the Query Designer](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. Design the stored procedure in the **Diagram** and **Grid** panes of the Query Designer.
4. Click **Save**  on the toolbar.
5. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules.

Microsoft Access adds the stored procedure to **Queries**  under **Objects** in the Database window.

[Create a multistatement stored procedure in the SQL Text Editor](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Create Text Stored Procedure**.

Access displays the SQL Text Editor with the following skeleton SQL syntax.

```
CREATE PROCEDURE "StoredProcedure1"
/*
    (
        @parameter1 datatype = default value,
        @parameter2 datatype OUTPUT
    )
*/
AS
    /* SET NOCOUNT ON */
    RETURN
```

3. Write the stored procedure by using Transact-SQL. For more information on Transact-SQL, see the SQL Server documentation.

Notes

- If you're creating a stored procedure that returns rows from the database, the first statement after the CREATE PROCEDURE statement must be a SELECT statement, or you must use SET NOCOUNT ON at the top of the procedure. This ensures that the result set from the first SELECT statement is displayed; otherwise, no rows will display. For more information on the SET NOCOUNT ON SQL statement, see the SQL Server documentation.
- You can graphically edit a SQL SELECT statement in a multistatement stored procedure by using the Query Builder.

[Create a user-defined function](#)

[Create an in-line user-defined function](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design In-Line Function**.
3. Design the in-line user-defined function in the **Diagram** and **Grid** panes of the Query Designer.
4. Click **Save**  on the toolbar.

5. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules.

Microsoft Access adds the in-line user-defined function to **Queries**  under **Objects** in the Database window.

[Create a table user-defined function](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Create Text Table-Valued Function**.

Access displays the SQL Text Editor with the following skeleton SQL syntax:

```
CREATE FUNCTION "Function1"
(
    /*
    @parameter1 datatype = default value,
    @parameter2 datatype
    */
)
RETURNS /* @table_variable TABLE (column1 datatype, column2 data
AS
BEGIN
    /* INSERT INTO @table_variable
       sql select statement */
    /* alternative sql statement or statements */
RETURN
END
```

Write the table user-defined function by using Transact-SQL. For more information on Transact-SQL, see the SQL Server documentation.

3. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules.

Microsoft Access adds the table user-defined function to **Queries**  under **Objects** in the Database window.

Note You can graphically edit an SQL SELECT statement in a table user-defined function by using the Query Builder.

[Create a scalar user-defined function](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Create Text Scalar Function**.

Access displays the SQL text editor with the following skeleton SQL syntax.

```
CREATE FUNCTION "Function1"  
(  
  /*  
  @parameter1 datatype = default value,  
  @parameter2 datatype  
  */  
)  
RETURNS /* datatype */  
AS  
  BEGIN  
    /* sql statement ... */  
  RETURN /* value */  
  END
```

Write the scalar user-defined function by using Transact-SQL. For more information on Transact-SQL, see the SQL Server documentation.

3. In the **Save As** dialog box, enter a name that follows Microsoft Access object-naming rules.

Microsoft Access adds the scalar user-defined function to **Queries**  under **Objects** in the Database window.

Note You can graphically edit an SQL SELECT statement in a scalar user-defined function by using the Query Builder.

Note You cannot use a user-defined function as the record source of a [data access page](#).

[Create an action query](#)

[Create an Append Values query](#)

Caution You cannot undo the action of an Append Values query. As a precaution, back up your data before executing the query.

[Background information](#)

You can create a new row in the current table using an Append Values query. When you create an Append Values query, you specify:

- The database table to add the row to.
- The columns whose contents you want to add.
- The value or expression to insert into the individual columns.

For example, the following query adds a row to the `titles` table, specifying values for the title, type, publisher, and price:

```
INSERT INTO titles
(title_id, title, type, pub_id, price)
VALUES ('BU9876', 'Creating Web Pages', 'business', '1389', '29.99')
```

When you create an Append Values query, the Grid pane changes to reflect the only options available for inserting a new row: the column name and the value to insert.

[Create an Append Values query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the table, view, or function you want to append values to, and then click **Close**.
4. On the **Query** menu click **Append Values Query**.

Note If more than one table, view, or function is displayed in the Grid pane when you start the Append Values query, the Query Designer displays the **Choose Target Table for Insert Values** dialog box to prompt you for the name of the table to update.

5. Define the data columns to create by adding them to the query. Columns will be updated only if you add them to the query.

6. In the **New Value** column of the Grid pane, enter the new value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

Caution The Query Designer cannot check that a value fits within the length of the column you are inserting. If you provide a value that is too long, it might be truncated without warning. For example, if a name column is 20 characters long but you specify an insert value of 25 characters, the last 5 characters might be truncated.

When you execute an Append Values query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were changed.

 [Create an Append query](#)

Caution You cannot undo the action of executing an Append query. As a precaution, back up your data before executing the query.

 [Background information](#)

You can copy rows from one table to another or within a table using an Append query. For example, in a titles table, you can use an Append query to copy information about all the titles for one publisher to a second table that you can make available to that publisher. An Append query is similar to a Make-Table query, but copies rows into an existing table.

When you create an Append query, you specify:

- The database table to copy rows to (the destination table).
- The table or tables to copy rows from (the source table). The source table or tables become part of a subquery. If you are copying within a table, the source table is the same as the destination table.
- The columns in the source table whose contents you want to copy.
- The target columns in the destination table to copy the data to.
- Search conditions to define the rows you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Group By options, if you want to copy only summary information.

For example, the following query copies title information from the `titles` table to an archive table called `archivetitles`. The query copies the contents of four columns for all titles belonging to a particular publisher:

```
INSERT INTO archivetitles
(title_id, title, type, pub_id)
SELECT title_id, title, type, pub_id
FROM titles
WHERE (pub_id = '0766')
```

Note To insert values into a new row, use an Append Values query.

You can copy the contents of selected columns or of all columns in a row. In either case, the data you are copying must be compatible with the columns in the rows you are copying to. For example, if you copy the contents of a column such as price, the column in the row you are copying to must accept numeric data with decimal places. If you are copying an entire row, the destination table must have compatible columns in the same physical position as the source table.

When you create an Append query, the Grid pane changes to reflect options available for copying data. Because you do not display data in an Append query, the Output column is removed. An Append column is added to allow you to specify the columns into which data should be copied.

 [Create an Append query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use to create the Append query, and then click **Close**.
4. On the **Query** menu click **Append Query**.

Note The Query Designer cannot determine in advance which tables and views you can update. Therefore, the **Table Name** list in the **Choose Table for Insert Results** dialog box shows all available tables and views in the data connection you are querying, even those that you might not be able to copy rows to.

5. Add to the query the table to copy rows from (the source table). If you are

copying rows within a table, you can add the source table as a destination table.

6. In the rectangle representing the table, view, or function, choose the names of the columns whose contents you want to copy. To copy entire rows, choose * **(All Columns)**.

The Query Designer adds the columns you choose to the **Column** column of the **Grid** pane.

7. In the **Append** column of the **Grid** pane, select a target column in the destination table for each column you are copying. Choose *tablename.** if you are copying entire rows. The columns in the destination table must have the same (or compatible) data types as the columns in the source table.
8. If you want to copy rows in a particular order, specify a sort order.
9. Specify the rows to copy by entering search conditions in the **Criteria** column.

If you do not specify a search condition, all rows will be copied to the destination table.

Note When you add a column to search to the **Grid** pane, the Query Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table, view, or function (+).

10. If you want to copy summary information, specify Group By options.

When you execute an Append query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were copied.

[Create an Update query](#)

Caution You cannot undo the action of executing an Update query. As a precaution, back up your data before executing the query.

[Background information](#)

You can change the contents of multiple rows in one operation by using an Update query. For example, in a `titles` table you can use an Update query to

add 10% to the price of all books for a particular publisher.

When you create an Update query, you specify:

- The table to update.
- The columns whose contents you want to update.
- The value or expression to use to update the individual columns.
- Search conditions to define the rows you want to update.

For example, the following query updates the titles table by adding 10% to the price of all titles for one publisher:

```
UPDATE titles
SET price = price * 1.1
WHERE (pub_id = '0766')
```



[Create an Update query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the table, view, or function you want to use to update, and then click **Close**.
4. On the **Query** menu click **Append Query**.

Note If more than one table is displayed in the Grid pane when you start the Update query, the Query Designer displays the **Update Table** dialog box to prompt you for the name of the table to update.

5. Define the data columns to update by adding them to the query. Columns will be updated only if you add them to the query.
6. In the **New Value** column of the Grid pane, enter the update value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

Caution The Query Designer cannot check that a value fits within the length of the column you are updating. If you provide a value that is too long, it might be truncated without warning. For example, if a name column

is 20 characters long but you specify an update value of 25 characters, the last 5 characters might be truncated.

7. Define the rows to update by entering search conditions in the **Criteria** column.

If you do not specify a search condition, all rows will be updated.

Note When you add a column to the Grid pane for use in a search condition, the Query Designer also adds it to the list of columns to be updated. If you want to use a column for a search condition but not update it, clear the check box next to the column name in the rectangle representing the table, view, or function . ().

When you execute an Update query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were changed.

 [Create a Delete query](#)

Caution You cannot undo the action of executing a Delete query. As a precaution, back up your data before executing a Delete query.

 [Background information](#)

You can delete multiple rows in one operation by using a Delete query. When you create a Delete query, you specify the database table to delete rows from and the search condition to define the rows you want to delete.

Note Deleting all rows from a table clears the data in the table but does not delete the table itself. You can delete tables using the Database Designer.

When you create a Delete query, the Grid pane changes to reflect the options available for deleting rows. Because you do not display data in a Delete query, the Output, Sort By, and Sort Order columns are removed. In addition, the check boxes next to the column names in the rectangle representing the table, view, or function are removed because you cannot specify individual columns to delete.

 [Create a delete query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use in the delete query, and then click **Close**.
4. On the **Query** menu click **Delete Query**.
5. Enter search conditions to define the rows to be deleted. If you do not specify a search condition, all rows will be deleted.

Note Because you cannot delete the contents of individual columns, the check boxes in the rectangle representing the table, view, or function are removed when you start a Delete query. To add columns to the Grid pane to use in search conditions, drag them from the rectangle representing the table, view, or function to the **Columns** column, or select a data column in the **Columns** column.

When you execute the Delete query by using the **Run** button on the Query Designer toolbar, a message appears indicating how many rows were deleted.

[Create a Make-Table query](#)

[Background information](#)

You can copy rows into a new table using a Make-Table query, which is useful for creating subsets of data to work with or copying the contents of a table from one database to another. A Make-Table query is similar to an Append query, but creates a new table to copy rows into.

When you create a Make-Table query, you specify:

- The name of the new database table (the destination table).
- The table or tables to copy rows from (the source table). You can copy from a single table or from joined tables.
- The columns in the source table whose contents you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Search conditions to define the rows you want to copy.
- Group By options, if you want to copy only summary information.

For example, the following query creates a new table called `uk_customers` and copies information from the `customers` table to it:

```
SELECT *
INTO uk_customers
FROM customers
WHERE country = 'UK'
```

In order to use a Make-Table query successfully, you must have permission to create a table in the target database.

[Create a Make-Table query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use to create the query, and then click **Close**.
4. On the **Query** menu click **Make- Table Query**.
5. In the **Make-Table** dialog box, type the name of the destination table. The Query Designer does not check whether the name is already in use or whether you have permission to create the table.

To create a destination table in another database, specify a fully qualified table name including the name of the target database, the owner (if required), and the name of the table using the following syntax:

```
database.owner.table
```

For more information on qualified object names, see the SQL Server documentation.

6. Specify the columns to copy by adding them to the query. Columns will be copied only if you add them to the query. To copy entire rows, choose ***** (**All Columns**).

The Query Designer adds the columns you choose to the **Column** column of the **Grid** pane.

7. If you want to copy rows in a particular order, specify a sort order.

8. Specify the rows to copy by entering search conditions.

If you do not specify a search condition, all rows will be copied to the destination table.

Note When you add a column to search to the **Grid** pane, the Query Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table, view, or function (▼).

9. If you want to copy summary information, specify **Group By** options.

When you execute a Make-Table query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were copied.

[Graphically design a SQL SELECT statement in the Query Builder.](#)

[Graphically design a SQL SELECT Statement in the record source of a form or report](#)

1. Open a form or report in [Design view](#)
2. Double-click the [form selector](#) or the [report selector](#) to open the property sheet for the form or report.
3. Click the **Build** button  next to the **RecordSource** property box to display the Query Builder.
4. Design the SQL SELECT statement in the **Diagram** and **Grid** panes of the Query Designer.
5. To save the SQL SELECT statement, close the Query Builder, and then close the property sheet.

Note Access does not add the SQL SELECT statement to **Queries**  under **Objects** in the [Database window](#).

[Graphically design a SQL SELECT statement in the SQL Text Editor](#)

1. In the [Database window](#), click **Queries**  under **Objects**, select a multistatement stored procedure, scalar user-defined function, or table user-defined function, and then click **Design**.

Access displays the multistatement stored procedure, scalar user-defined function, or table user-defined function in the SQL text editor.

2. Do one of the following:

[Edit an existing SQL SELECT statement](#)

1. Right click on the SQL SELECT statement, and then click **Edit SQL**.

Microsoft Access prompts you to load the SQL SELECT statement in the Query Builder.

2. Click **Yes**.

[Create a new SQL SELECT statement](#)

o Right click and select **Insert SQL**.

Access opens the Query Builder with a blank **Diagram** and **Grid** panes.

3. Make the changes you want in the Query Builder.

Note Access does not automatically verify the syntax of the SQL statement. You can use the **Verify SQL Syntax**  button on the Query Builder toolbar.

4. Close the Query Builder.

Access inserts the new or revised SQL SELECT statement into the SQL Text Editor.

[Show All](#)

Modify an indexed view (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Note Indexed views are supported when your [Microsoft Access project](#) is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

There are two ways you modify an indexed view. You can modify the view, or you can modify the indexes on the view.

[Modify an indexed view](#)

1. In the Database window, click **Queries**  under **Objects**, click the indexed view you want to open, and then click **Design** on the database window toolbar.
2. Modify the view in the **Diagram** and **Grid** panes of the Query Designer.

Note When you save the modified view, its indexes will be deleted and recreated.

[Modify the index of an indexed view](#)

1. In the Database window, click **Queries**  under **Objects**, click the indexed view you want to open, and then click **Design** on the database window toolbar.
2. Right-click the background of the Diagram pane and choose **Manage Indexes** from the shortcut menu.
3. Revise the information for the index definition— index name, index columns and their order, index file group, and the other index settings. For

more information about each dialog box option, click **Help**.

[Show All](#)

Create and use a parameter query (ADP)

 [Create a parameter query by using an existing stored procedure or inline user-defined function](#)

1. In the [Database window](#), under **Objects**, click **Queries** , click the stored procedure or inline user-defined function you want to open, and then click **Design** on the database window toolbar.
2. Do one or more of the following:
 - In the **Criteria** cell for each column you want to prompt a value for, type a named parameter with no embedded spaces preceded by an at sign (@) as the following example shows:

@Enter_Customer

- In the **Column** cell, enter a named parameter in a computed column expression as the following example shows:

Price * @Enter_Discount

3. To change the data type of the parameter, right-click in the **Design** grid of the Query Designer, select **Properties** from the shortcut menu, and then click the **Stored Procedure Parameters** or **Function Parameters** tab.

Change the [data type](#) when you want to do a calculation and you must ensure that the parameter is a number data type, such as [int](#) or [money](#). If you type a text data type, use the short form to specify the text length, for example [nvarchar\(50\)](#) or [char\(100\)](#).

4. To change the parameter default, right-click in the Query Designer, select **Properties** from the shortcut menu, and then click the **Stored Procedure Parameters** or **Function Parameters** tab. The value you enter will be entered as the default value. For example, if you enter a parameter default value of 'CA', when you run a parameter query with the following SQL statement:

```
SELECT * from authors WHERE state = @state
```

The parameter query will enter CA as the parameter default value and not prompt you for a parameter.

5. Click **Save**  to save the stored procedure or inline user-defined function.
6. Run the parameter query to verify that it works as intended.

[How?](#)

1. Do one of the following:
 - In the Database window, click **Queries**  under **Objects**, click the [parameter query](#) you want to run, and then click **Open** on the database window toolbar.
 - In the Query Designer, Click **Run**  to run the parameter query.

Access shows the **Enter Parameter Value** dialog box for each parameter in the query.

2. Do one of the following for each parameter value you are prompted for:
 - To enter a parameter value, type the value.
 - To enter the [default value](#), select <DEFAULT>.
 - To enter a [Null](#) value, select <NULL>.
 - To enter a [zero-length](#) or blank string, leave the box blank.

Note You cannot run or create a [view](#) as a parameter query.

[Use a stored procedure or inline user-defined function parameter query as the record source of a form or report](#)

When you use a stored procedure or inline user-defined function parameter query as the record source of a form or report, you can do one of two things:

- Pass the value of the parameter to a field control on the form or report
- Prompt for the value of the parameter in the **Enter Parameter Value** dialog box.

To do this, you use the **InputParameters** property on the form or report, which maps the named parameters in the stored procedure or inline user-defined function to expressions that reference the field control in the form or report.

For example if you have a parameter query that uses the following named parameters:

```
SELECT State, SalesYear from Orders WHERE Orders.State =  
@EnterState AND Orders.SalesYear =@EnterSalesYear
```

You can set the **InputParameters** property as follows:

```
@State char=[Form]![StateList], @SalesYear smallint=[Enter year of  
interest]
```

For the first parameter, Access sets the *@State* parameter to be the current value of the StateList control. For the second parameter, Access prompts the user for the *@SalesYear* parameter. If there were any other parameters to the stored procedure or inline user-defined function that were not in this list, they would have default values assigned.

 [Create and run a parameter query in the SQL SELECT statement of a form or report](#)

1. In the Database window, under **Objects**, click **Queries** , click the form or report you want to open, and then click **Design** on the database window toolbar.
2. In the Database window, click the form or report that contains the SQL statement you want to add parameters to, and open it in [Design view](#).
3. Double-click the [form selector](#) or the [report selector](#) to open the property sheet.
4. Click the **Build** button  next to the **RecordSource** property to open the [query](#).
5. Add the columns or [expressions](#) that you want to search to the **Grid** pane.
6. Locate the row containing the data column or expression to search, and then, in the **Criteria** column, enter one or more unnamed parameters by typing a question mark (?).

When you use an SQL SELECT statement parameter query as the record source of a form or report, you can do one of two things:

- Pass the value of the parameter to a field control on the form or report
- Prompt for the value of the parameter in the **Enter Parameter Value** dialog box.

To do this, you use the **InputParameters** property on the form or report, which maps the unnamed parameters in the SQL SELECT statement to expressions that reference the field control in the form or report.

For example, if you have an SQL SELECT statement parameter query that uses the following unnamed parameters:

```
SELECT State, SalesYear from Orders WHERE Orders.State = ? AND  
Orders.SalesYear = ?
```

You can set the **InputParameters** property as follows:

```
? char=[Form]![StateList], @SalesYear smallint=[Enter year of  
interest]
```

For the first parameter, Access sets the *@State* parameter to be the current value of the StateList control. For the second parameter, Access prompts the user for the *@SalesYear* parameter. If there were any other parameters in the SQL SELECT statement, they would have default values assigned.

[Show All](#)

Execute a query in the Query Designer (ADP)

When you have finished designing your query, you can run it.

1. If you want to test whether the syntax of the query you are creating is correct, you can verify the query.

[How?](#)

To avoid problems, you can check the query you have built to ensure its syntax is correct. This option is especially useful when you enter statements in the SQL pane.

Note A statement can be valid, and therefore be verified successfully, even if it cannot be represented in the Diagram and Grid panes.

Note SQL Verification can detect some, but not all SQL errors. If a query contains an error not detected during SQL verification, the database will detect the error when you run the query.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Click **Verify SQL Syntax** .
2. Click **Run** .

Notes

- If you are creating a Select query, the results of the query appear in datasheet view.
- If you are creating an Update, Append, Append Values, Delete, or Make-Table query, the Query Designer displays a message indicating how many rows were affected by the query.

- To stop a query that is taking too long to finish, press CTRL+BREAK.

[Show All](#)

Create a make-table query (ADP)

[Background information](#)

You can copy rows into a new table using a Make-Table query, which is useful for creating subsets of data to work with or copying the contents of a table from one database to another. A Make-Table query is similar to an Append query, but creates a new table to copy rows into.

When you create a Make-Table query, you specify:

- The name of the new database table (the destination table).
- The table or tables to copy rows from (the source table). You can copy from a single table or from joined tables.
- The columns in the source table whose contents you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Search conditions to define the rows you want to copy.
- Group By options, if you want to copy only summary information.

For example, the following query creates a new table called `uk_customers` and copies information from the `customers` table to it:

```
SELECT *  
INTO uk_customers  
FROM customers  
WHERE country = 'UK'
```

In order to use a Make-Table query successfully, you must have permission to create a table in the target database.

[Create a Make-Table query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use to create the query, and then click **Close**.

4. On the **Query** menu click **Make- Table Query**.
5. In the **Make-Table** dialog box, type the name of the destination table. The Query Designer does not check whether the name is already in use or whether you have permission to create the table.

To create a destination table in another database, specify a fully qualified table name including the name of the target database, the owner (if required), and the name of the table using the following syntax:

```
database.owner.table
```

For more information on qualified object names, see the SQL Server documentation.

6. Specify the columns to copy by adding them to the query. Columns will be copied only if you add them to the query. To copy entire rows, choose *** (All Columns)**.

The Query Designer adds the columns you choose to the **Column** column of the **Grid** pane.

7. If you want to copy rows in a particular order, specify a sort order.
8. Specify the rows to copy by entering search conditions.

If you do not specify a search condition, all rows will be copied to the destination table.

Note When you add a column to search to the **Grid** pane, the Query Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table, view, or function (✓).

9. If you want to copy summary information, specify **Group By** options.

When you execute a Make-Table query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were copied.

[Show All](#)

Create an append query (ADP)

Caution You cannot undo the action of executing an Append query. As a precaution, back up your data before executing the query.



[Background information](#)

You can copy rows from one table to another or within a table using an Append query. For example, in a titles table, you can use an Append query to copy information about all the titles for one publisher to a second table that you can make available to that publisher. An Append query is similar to a Make-Table query, but copies rows into an existing table.

When you create an Append query, you specify:

- The database table to copy rows to (the destination table).
- The table or tables to copy rows from (the source table). The source table or tables become part of a subquery. If you are copying within a table, the source table is the same as the destination table.
- The columns in the source table whose contents you want to copy.
- The target columns in the destination table to copy the data to.
- Search conditions to define the rows you want to copy.
- Sort order, if you want to copy the rows in a particular order.
- Group By options, if you want to copy only summary information.

For example, the following query copies title information from the titles table to an archive table called archivetimes. The query copies the contents of four columns for all titles belonging to a particular publisher:

```
INSERT INTO archivetimes
(title_id, title, type, pub_id)
SELECT title_id, title, type, pub_id
FROM titles
WHERE (pub_id = '0766')
```

Note To insert values into a new row, use an Append Values query.

You can copy the contents of selected columns or of all columns in a row. In either case, the data you are copying must be compatible with the columns in the rows you are copying to. For example, if you copy the contents of a column such as price, the column in the row you are copying to must accept numeric data with decimal places. If you are copying an entire row, the destination table must have compatible columns in the same physical position as the source table.

When you create an Append query, the Grid pane changes to reflect options available for copying data. Because you do not display data in an Append query, the Output column is removed. An Append column is added to allow you to specify the columns into which data should be copied.

[Create an Append query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use to create the Append query, and then click **Close**.
4. On the **Query** menu click **Append Query**.

Note The Query Designer cannot determine in advance which tables and views you can update. Therefore, the **Table Name** list in the **Choose Table for Insert Results** dialog box shows all available tables and views in the data connection you are querying, even those that you might not be able to copy rows to.

5. Add to the query the table to copy rows from (the source table). If you are copying rows within a table, you can add the source table as a destination table.
6. In the rectangle representing the table, view, or function, choose the names of the columns whose contents you want to copy. To copy entire rows, choose ***** (**All Columns**).

The Query Designer adds the columns you choose to the **Column** column of the **Grid** pane.

7. In the **Append** column of the **Grid** pane, select a target column in the destination table for each column you are copying. Choose *tablename.** if

you are copying entire rows. The columns in the destination table must have the same (or compatible) data types as the columns in the source table.

8. If you want to copy rows in a particular order, specify a sort order.
9. Specify the rows to copy by entering search conditions in the **Criteria** column.

If you do not specify a search condition, all rows will be copied to the destination table.

Note When you add a column to search to the **Grid** pane, the Query Designer also adds it to the list of columns to copy. If you want to use a column for searching but not copy it, clear the check box next to the column name in the rectangle representing the table, view, or function (+).

10. If you want to copy summary information, specify Group By options.

When you execute an Append query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were copied.

[Show All](#)

Create an update query (ADP)

Caution You cannot undo the action of executing an Update query. As a precaution, back up your data before executing the query.

[Background information](#)

You can change the contents of multiple rows in one operation by using an Update query. For example, in a `titles` table you can use an Update query to add 10% to the price of all books for a particular publisher.

When you create an Update query, you specify:

- The table to update.
- The columns whose contents you want to update.
- The value or expression to use to update the individual columns.
- Search conditions to define the rows you want to update.

For example, the following query updates the `titles` table by adding 10% to the price of all titles for one publisher:

```
UPDATE titles
SET price = price * 1.1
WHERE (pub_id = '0766')
```

[Create an Update query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the table, view, or function you want to use to update, and then click **Close**.
4. On the **Query** menu click **Append Query**.

Note If more than one table is displayed in the Grid pane when you start the Update query, the Query Designer displays the **Update Table** dialog

box to prompt you for the name of the table to update.

5. Define the data columns to update by adding them to the query. Columns will be updated only if you add them to the query.
6. In the **New Value** column of the Grid pane, enter the update value for the column. You can enter literal values, column names, or expressions. The value must match (or be compatible with) the data type of the column you are updating.

Caution The Query Designer cannot check that a value fits within the length of the column you are updating. If you provide a value that is too long, it might be truncated without warning. For example, if a name column is 20 characters long but you specify an update value of 25 characters, the last 5 characters might be truncated.

7. Define the rows to update by entering search conditions in the **Criteria** column.

If you do not specify a search condition, all rows will be updated.

Note When you add a column to the Grid pane for use in a search condition, the Query Designer also adds it to the list of columns to be updated. If you want to use a column for a search condition but not update it, clear the check box next to the column name in the rectangle representing the table, view, or function . ().

When you execute an Update query by using the **Run**  button on the Query Designer toolbar, a message appears indicating how many rows were changed.

[Show All](#)

Create a delete query (ADP)

Caution You cannot undo the action of executing a Delete query. As a precaution, back up your data before executing a Delete query.

[Background information](#)

You can delete multiple rows in one operation by using a Delete query. When you create a Delete query, you specify the database table to delete rows from and the search condition to define the rows you want to delete.

Note Deleting all rows from a table clears the data in the table but does not delete the table itself. You can delete tables using the Database Designer.

When you create a Delete query, the Grid pane changes to reflect the options available for deleting rows. Because you do not display data in a Delete query, the Output, Sort By, and Sort Order columns are removed. In addition, the check boxes next to the column names in the rectangle representing the table, view, or function are removed because you cannot specify individual columns to delete.

[Create a delete query](#)

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New**.
2. In the **New Query** dialog box, select **Design Stored Procedure**.
3. In the **Add Table** dialog box, add the tables, views, or functions you want to use in the delete query, and then click **Close**.
4. On the **Query** menu click **Delete Query**.
5. Enter search conditions to define the rows to be deleted. If you do not specify a search condition, all rows will be deleted.

Note Because you cannot delete the contents of individual columns, the check boxes in the rectangle representing the table, view, or function are removed when you start a Delete query. To add columns to the Grid pane to use in search conditions, drag them from the rectangle representing the table, view, or function to the **Columns** column, or select a data column in

the **Columns** column.

When you execute the Delete query by using the **Run** button on the Query Designer toolbar, a message appears indicating how many rows were deleted.

[Show All](#)

Rules and grid-pane conventions for combining search conditions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can create queries that include any number of search conditions, linked with any number of AND and OR operators. A query with a combination of AND and OR clauses can become complex, so it is helpful to understand how such a query is interpreted when you execute it, and how such a query is represented in the Grid and SQL panes.

To help you understand the discussion below, imagine that you are working with an employee table containing the columns `hire_date`, `job_lvl`, and `status`. The examples assume that you need to know information such as how long an employee has worked with the company (that is, what the employee's hire date is), what type of job the employee performs (what the job level is), and the employee's status (for example, retired).

[Precedence of AND and OR](#)

When a query is executed, it evaluates first the clauses linked with AND, and then those linked with OR.

Note The NOT operator takes precedence over both AND and OR.

For example, to find either employees who have been with the company for more than five years in lower-level jobs or employees with middle-level jobs without regard for their hire date, you can construct a WHERE clause such as the following:

```
WHERE
  hire_date < '01/01/90' AND
  job_lvl = 100 OR
  job_lvl = 200
```

To override the default precedence of AND over OR, you can put parentheses around specific conditions in the SQL pane. Conditions in parentheses are always evaluated first. For example, to find all employees who have been with the company more than five years in either lower or middle-level jobs, you can construct a WHERE clause such as the following:

```
WHERE  
  hire_date < '01/01/90' AND  
  (job_lvl = 100 OR job_lvl = 200)
```

Tip

It is recommended that, for clarity, you always include parentheses when combining AND and OR clauses instead of relying on the default precedence.

[How AND works with multiple OR clauses](#)

Understanding how AND and OR clauses are related when combined can help you construct and understand complex queries in the Query Designer.

If you link multiple conditions using AND, the first set of conditions linked with AND applies to all the conditions in the second set. In other words, a condition linked with AND to another condition is distributed to all the conditions in the second set. For example, the following schematic representation shows an AND condition linked to a set of OR conditions:

A AND (B OR C)

The representation above is logically equivalent to the following schematic representation, which shows how the AND condition is distributed to the second set of conditions:

(A AND B) OR (A AND C)

This distributive principle affects how you use the Query Designer. For example, imagine that you are looking for all employees who have been with the company more than five years in either lower or middle-level jobs. You enter the following WHERE clause into the statement in the SQL pane:

```
WHERE (hire_date < '01/01/90' ) AND  
      (job_lvl = 100 OR job_lvl = 200)
```

The clause linked with AND applies to both clauses linked with OR. An explicit way to express this is to repeat the AND condition once for each condition in the OR clause. The following statement is more explicit (and longer) than the previous statement, but is logically equivalent to it:

```
WHERE (hire_date < '01/01/90' ) AND  
      (job_lvl = 100) OR  
      (hire_date < '01/01/90' ) AND  
      (job_lvl = 200)
```

The principle of distributing AND clauses to linked OR clauses applies no matter how many individual conditions are involved. For example, imagine that you want to find higher or middle-level employees who have been with the company more than five years or are retired. The WHERE clause might look like this:

```
WHERE  
      (job_lvl = 200 OR job_lvl = 300) AND  
      (hire_date < '01/01/90' ) OR (status = 'R')
```

After the conditions linked with AND have been distributed, the WHERE clause will look like this:

```
WHERE  
      (job_lvl = 200 AND hire_date < '01/01/90' ) OR  
      (job_lvl = 200 AND status = 'R') OR  
      (job_lvl = 300 AND hire_date < '01/01/90' ) OR  
      (job_lvl = 300 AND status = 'R')
```



[How Multiple AND and OR Clauses are represented in the Grid pane](#)

The Query Designer represents your search conditions in the Grid pane. However, in some cases that involve multiple clauses linked with AND and OR, the representation in the Grid pane might not be what you expect. In addition, if you modify your query in the Grid or Diagram panes, you might find that your SQL statement has been changed from what you entered.

In general, these rules dictate how AND and OR clauses appear in the Grid pane:

- All conditions linked with AND appear in the **Criteria** grid column or in the same **Or ...** column.
- All conditions linked with OR appear in separate **Or ...** columns.
- If the logical result of a combination of AND and OR clauses is that the AND is distributed into several OR clauses, the Grid pane represents this explicitly by repeating the AND clause as many times as necessary.

For example, in the SQL pane you might create a search condition such as the following, in which two clauses linked with AND take precedence over a third one linked with OR:

```
WHERE (hire_date < '01/01/90' ) AND
      (job_lvl = 100) OR
      (status = 'R')
```

The Query Designer represents this WHERE clause in the Grid pane as follows:

Column	Criteria	Or...
hire_date	< '1/1/90'	
job_lvl	= 100	
status		= 'R'

However, if the linked OR clauses take precedence over an AND clause, the AND clause is repeated for each OR clause. This causes the AND clause to be distributed to each OR clause. For example, in the SQL pane you might create a WHERE clause such as the following:

```
WHERE (hire_date < '01/01/90' ) AND
      ( (job_lvl = 100) OR
      (status = 'R') )
```

The Query Designer represents this WHERE clause in the Grid pane as follows:

Column	Criteria	Or...
hire_date	< '1/1/90'	< '1/1/90'
job_lvl	= 100	
status		= 'R'

If the linked OR clauses involve only one data column, the Query Designer can

place the entire OR clause into a single cell of the grid, avoiding the need to repeat the AND clause. For example, in the SQL pane you might create a WHERE clause such as the following:

```
WHERE (hire_date < '01/01/90' ) AND  
      ((status = 'R') OR (status = 'A'))
```

The Query Designer represents this WHERE clause in the Grid pane as follows:

Column	Criteria
hire_date	< '1/1/90'
status	= 'R' OR = 'A'

If you make a change to the query (such as changing one of the values in the Grid pane), the Query Designer recreates the SQL statement in the SQL pane. The recreated SQL statement will resemble the Grid pane display rather than your original statement. For example, if the Grid pane contains distributed AND clauses, the resulting statement in the SQL pane will be recreated with explicit distributed AND clauses.

[Show All](#)

Comparison operators (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Overview of search condition operators](#)

You can use any standard SQL operators in a search condition. When you use operators in a search condition, the following rules apply:

- The data types of the data used in the comparison must match. That is, only text can be compared to text, numbers to numbers, and so on. You can use a function or keyword (such as CAST) to temporarily convert data to another data type.
- If you compare text data, the result depends on the character set currently in use. For example, if a table was created using Scandinavian characters, the search results might differ depending on whether your current character set (code page) is Scandinavian or another character set.
- If a comparison value is null, the result is unknown. Nulls are not matched to any value, including other instances of null. For example, if you are searching for a name beginning with the letter "M" or higher (name >= 'M'), and some of the rows contain no value, those rows do not appear, no matter what comparison operator you use.

 [Summary of search condition operators](#)

The following table summarizes search condition operators that are defined for standard SQL.

Operator	Meaning	Grid pane example
=	Equal.	= 'Smith'

<> Not equal <> 'Active'
!= to.

> Greater > '01 Jan 1995'²
than.

< Less than. < 100

>= Greater >= 'T'
!< than or
equal to.

<= Less than <= '01 Jan 1995'²
!> or equal to.

BETWEEN BETWEEN
expr1 Tests range '01 Jan 1995'
AND AND
AND *expr2* of values. '31 Dec 1995'²

IS [NOT] Tests
NULL whether
contents of IS NULL
column or
result of
expression
is null.
Performs

[NOT] LIKE	pattern matching (usually restricted to character data types).	LIKE ('MAC%')
<i>expr1</i> [NOT] IN (<i>val1</i> , <i>val2</i> , ...)	Matches list of specific values by testing whether	IN ('SW', 'SE')
– or –	<i>expr1</i> appears either in a	supplier_id IN (subquery)
<i>expr1</i> [NOT] IN (<i>subquery</i>)	list of values or in the result set of a subquery. Tests whether one or more rows in the result set of a subquery meet the specified condition.	
ANY (SOME)	(ANY and SOME are synonyms; the Query	<> ANY (subquery)

Designer
will use
ANY when
creating an
SQL
statement.)

ALL Tests
whether all
rows in the
result set
of a advance > ALL (subquery)
subquery
meet the
specified
condition.

[NOT]
EXISTS Tests
whether a
subquery
returns any
results EXISTS (subquery)
whatsoever
(not a
specific
result).

Logical operators (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can combine or modify search conditions using the standard logical operators listed in the following table. The operators are listed in the order that they are evaluated.

Operator	Meaning	Example
NOT	Logical opposite of condition	<code>SELECT * FROM employee WHERE NOT (fname = 'Ann')</code>
AND	Both conditions must apply	<code>SELECT * FROM employee WHERE lname = 'Smith' AND fname = 'Ann'</code>
OR	Either condition can apply	<code>SELECT * FROM employee WHERE region = 'UK' OR region = 'FRA'</code>

Select rows that do not match a value in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

To find rows that do not match a value, use the NOT operator. For example, to find all the rows in a products table where the values in the product code column begin with a character other than "A," you can enter a search condition such as the following:

```
NOT LIKE 'A%'
```

[Show All](#)

Rules for entering search values (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

This topic discusses the conventions you must use when entering the following types of literal values for a search condition:

 [Searching on text values](#)

The following guidelines apply when you enter text values in search conditions:

- **Quotation marks** Enclose text values in single quotation marks, as in this example for a last name:

'Smith'

If you are entering a search condition in the Grid pane, you can simply type the text value and the Query Designer will automatically put single quotation marks around it.

Note In Microsoft SQL Server, terms in single quotation marks are interpreted as literal values, whereas terms in double quotation marks are interpreted as database objects such as column or table references. Therefore, even though the Query Designer can accept terms in double quotation marks, it might interpret them differently than you expect.

- **Embedding apostrophes** If the data you are searching for contains a single quotation mark (an apostrophe), you can enter two single quotation marks to indicate that you mean the single quotation mark as a literal value and not a delimiter. For example, the following condition searches for the value "Swann's Way":

= 'Swann' 's Way'

- **Length limits** Do not exceed the maximum length of the SQL statement for your database when entering long strings.
- **Case-sensitivity** Follow the case-sensitivity rules for the database you are using. The database you are using determines whether text searches are case sensitive. Depending on how SQL Server was installed, some databases interpret the operator "=" to mean an exact case-sensitive match, but others will allow matches on any combination of uppercase and lowercase characters. For more information on case-sensitivity, see the SQL Server documentation.

If you are unsure about whether the database uses a case-sensitive search, you can use the UPPER or LOWER functions in the search condition to convert the case of the search data, as illustrated in the following example:

```
WHERE UPPER(lname) = 'SMITH'
```

[Searching on numeric values](#)

The following guidelines apply when you enter numeric values in search conditions:

- **Quotation marks** Do not enclose numbers in quotation marks.
- **Non-numeric characters** Do not include non-numeric characters except the decimal separator (as defined in the **Regional Settings** dialog box of Windows Control Panel) and negative sign (-). Do not include digit grouping symbols (such as a comma between thousands) or currency symbols.
- **Decimal marks** If you are entering whole numbers, you can include a decimal mark, whether the value you are searching for is an integer or a real number.
- **Scientific notation** You can enter very large or very small numbers using scientific notation, as in this example:

```
> 1.23456e-9
```

[Searching on dates](#)

The format you use to enter dates depends on the database you are using and in

what pane of the Query Designer you are entering the date. The Query Designer can work with the following date formats:

- **Locale-specific** The format specified for dates in the **Windows Regional Settings Properties** dialog box.
- **Database-specific** Any format understood by the database.
- **ANSI standard date** A format that uses braces, the marker 'd' to designate the date, and a date string, as in the following example:

```
{ d '1990-12-31' }
```

- **ANSI standard datetime** Similar to ANSI-standard date, but uses 'ts' instead of 'd' and adds hours, minutes, and seconds to the date (using a 24-hour clock), as in this example for December 31, 1990:

```
{ ts '1990-12-31 00:00:00' }
```

In general, the ANSI standard date format is used with databases that represent dates using a true date data type. In contrast, the datetime format is used with databases that support a datetime data type.

The following table summarizes the date format that you can use in different panes of the Query Designer.

Pane or view	Date format
Grid	Locale-specific Database-specific ANSI standard Dates entered in the Grid pane are converted to a database-compatible format in the SQL pane.
SQL	Database-specific ANSI standard Dates entered into the SQL pane are converted to the locale-specific format in the Grid pane.

Datasheet Locale-specific

[Searching on logical values](#)

A value of False is stored as zero (0). A value of True is stored as 1. The following guidelines apply when you enter logical values in search conditions:

- To search for a value of False, use a zero, as in the following example:

```
SELECT * FROM authors  
WHERE contract = 0
```

- If you are not sure what format to use when searching for a True value, try using 1, as in the following example:

```
SELECT * FROM authors  
WHERE contract = 1
```

- Alternatively, you can broaden the scope of the search by searching for any non-zero value, as in the following example:

```
SELECT * FROM authors  
WHERE contract <> 0
```

[Show All](#)

Specify search conditions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

You can specify the data rows that appear in your query by specifying search conditions. For example, if you are querying an employee table, you can specify that you want to find only the employees who work in a particular region.

You specify search conditions using an expression. Most commonly the expression consists of an operator and a search value. For example, to find employees in a particular sales region, you might specify the following search criterion for the region column:

= 'UK'

Note If you are working with multiple tables, the Query Designer examines each search condition to determine whether the comparison you are making results in a join. If so, the Query Designer automatically converts the search condition into a join.

[Specify search conditions](#)

1. If you have not done so already, add the columns or expressions that you want to use within your search condition to the Grid pane

If you are creating a Select query and do not want the search columns or expressions to appear in the query output, clear the **Output** column for each search column or expression to remove them as output columns.

2. Locate the row containing the data column or expression to search, and then in the **Criteria** grid column, enter a search condition.

Note If you do not enter an operator, the Query Designer automatically

inserts the equality operator "=".

The Query Designer updates the SQL statement in the SQL pane by adding or modifying the WHERE clause.

[Specify multiple search conditions for one column](#)

[Background information](#)

In some instances, you might want to apply a number of search conditions to the same data column. For example, you might want to:

- Search for several different names in an employee table or for employees who are in different salary ranges. This type of search requires an OR condition.
- Search for a book title that both starts with the word "The" and contains the word "Cook." This type of search requires an AND condition.

Note The information in this topic applies to search conditions in both the WHERE and HAVING clauses of a query. The examples focus on creating WHERE clauses, but the principles apply to both types of search conditions.

To search for alternative values in the same data column, you specify an OR condition. To search for values that meet several conditions, you specify an AND condition.

[Specify an OR Condition](#)

Using an OR condition enables you to specify several alternative values to search for in a column. This option expands the scope of the search and can return more rows than searching for a single value.

[Tip](#)

You can often use the IN operator instead to search for multiple values in the same data column.

1. In the Database window, click **Queries**  under **Objects**, click the query

- you want to open, and then click **Design** on the Database window toolbar.
2. In the Grid pane, add the column to search.
 3. In the **Criteria** column for the data column you just added, specify the first condition.
 4. In the **Or ...** column for the same data column, specify the second condition.

Column	Criteria	Or...
salary	< 30000	> 100000

The Query Designer creates a WHERE clause that contains an OR condition such as the following:

```
SELECT fname, lname
FROM employees
WHERE (salary < 30000) OR (salary > 100000)
```

[Specify an AND Condition](#)

Using an AND condition enables you to specify that values in a column must meet two (or more) conditions for the row to be included in the result set. This option narrows the scope of the search and usually returns fewer rows than searching for a single value.

[Tip](#)

If you are searching for a range of values, you can use the BETWEEN operator instead of linking two conditions with AND.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the Database window toolbar.
2. In the Grid pane, add the column to search.
3. In the **Criteria** column for the data column you just added, specify the first condition.
4. Add the same data column to the **Grid** pane again, placing it in an empty row of the grid.
5. In the **Criteria** column for the second instance of the data column, specify the second condition.

Column	Criteria
title	LIKE '%Cook%'
title	LIKE '%Recipe%'

The Query Designer creates a WHERE clause that contains an AND condition such as the following:

```
SELECT title_id, title
FROM titles
WHERE (title LIKE '%Cook%') AND
(title LIKE '%Recipe%')
```

[Specify multiple search conditions for multiple columns](#)

[Background information](#)

You can expand or narrow the scope of your query by including several data columns as part of your search condition. For example, you might want to:

- Search for employees who either have worked more than five years at the company or who hold certain jobs.
- Search for a book that is both published by a specific publisher and pertains to cooking.

To create a query that searches for values in either of two (or more) columns, you specify an OR condition. To create a query that must meet all conditions in two (or more) columns, you specify an AND condition.

[Specify an OR Condition](#)

To create multiple conditions linked with OR, you put each separate condition in a different column of the Grid pane.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the Database window toolbar.
2. In the **Grid** pane, add the columns you want to search.
3. In the **Criteria** column for the first column to search, specify the first condition.
4. In the **Or ...** column for the second data column to search, specify the

second condition, leaving the **Criteria** column blank.

Column	Criteria	Or...
job_lvl	>= 200	
hire_date		< '01 Jan 1990'

The Query Designer creates a WHERE clause that contains an OR condition such as the following:

```
SELECT job_lvl, hire_date
FROM employee
WHERE (job_lvl >= 200) OR
      (hire_date < '01/01/90')
```

5. Repeat Steps 3 and 4 for each additional condition you want to add. Use a different **Or ...** column for each new condition.

[Specify an AND Condition](#)

To search different data columns using conditions linked with AND, you put all the conditions in the **Criteria** column of the grid.

1. In the Grid pane, add the columns you want to search.
2. In the **Criteria** column for the first data column to search, specify the first condition.
3. In the **Criteria** column for the second data column, specify the second condition.

Column	Criteria	Or...
pub_id	= '0877'	
title	LIKE '%Cook%'	

The Query Designer creates a WHERE clause that contains an AND condition such as the following:

```
SELECT pub_id, title
FROM titles
WHERE (pub_id = '0877') AND (title LIKE '%Cook%')
```

4. Repeat Steps 2 and 3 for each additional condition you want to add.

[Combine conditions when OR has precedence](#)

To link conditions with OR and give them precedence over conditions linked with AND, you must repeat the AND condition for each OR condition.

For example, imagine that you want to find all employees who have been with the company more than five years and have lower-level jobs or are retired. This query requires three conditions, a single condition linked to two additional conditions with AND:

- Employees with a hire date earlier than five years ago, and
 - Employees with a job level of 100 or whose status is "R" (for retired).
1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the Database window toolbar.
 2. In the **Grid** pane, add the data columns you want to search. If you want to search the same column using two or more conditions linked with AND, you must add the data column name to the grid once for each value you want to search.
 3. Create the conditions to be linked with OR by entering the first one into the **Criteria** grid column and the second (and subsequent ones) into separate **Or ...** columns. For example, to link conditions with OR that search the job_lvl and status columns, enter values as shown here:

	Column	Table	Output	Criteria	Or...
	job_lvl	employee	<input type="checkbox"/>	= 100	
	status		<input type="checkbox"/>		= 'R'

Entering the values shown in the grid above produces the following WHERE clause in the statement in the SQL pane:

```
WHERE (job_lvl = 100) OR (status = 'R')
```

4. Create the AND condition by entering it once for each OR condition. Place each entry in the same grid column as the OR condition it corresponds to. For example, to add an AND condition that searches the hire_date column and applies to both OR conditions, enter values as shown here:

Column	Table	Output	Criteria	Or...
job_lvl	employee		= 100	
status				= 'R'
hire_date	employee		< '1/1/91'	< '1/1/91'

Entering the values shown in the grid above produces the following WHERE clause in the statement in the SQL pane:

```
WHERE (job_lvl = 100) AND
      (hire_date < '01/01/91' ) OR
      (status = 'R') AND
      (hire_date < '01/01/91' )
```

 [Tip](#)

You can repeat an AND condition by adding it once, and then using the **Cut** and **Paste** commands from the **Edit** menu to repeat it for other OR conditions.

The WHERE clause created by the Query Designer is equivalent to the following WHERE clause, which uses parentheses to specify the precedence of OR over AND:

```
WHERE (job_lvl = 100 OR status = 'R') AND
      (hire_date < '01/01/91')
```

Note If you enter the search conditions in the format shown immediately above in the SQL pane, but then make a change to the query in the Diagram or Grid panes, the Query Designer recreates the SQL statement to match the form with the AND condition explicitly distributed to both OR conditions.

 [Combine conditions when AND has precedence](#)

To combine conditions with AND, you put the conditions in the same column of the Grid pane. To combine conditions with OR, you put the first one in the Criteria column and additional conditions into an Or ... column.

For example, imagine that you want to find either employees who have been with the company for more than five years in lower-level jobs or employees with middle-level jobs regardless of their hire date. This query requires three

conditions, two of them linked with AND:

- Employees with a hire date earlier than five years ago and with a job level of 100

-or-

- Employees with a job level of 200

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the Database window toolbar.
2. In the **Grid** pane, add the data columns you want to search. If you want to search the same column using two or more conditions linked with AND, you must add the data column name to the grid once for each value you want to search.
3. In the **Criteria** column, enter all the conditions that you want to link with AND. For example, to link conditions with AND that search the hire_date and job_lvl columns, enter values as shown here:

	Column	Table	Output	Criteria	Or...
	hire_date	employee	<input type="checkbox"/>	< '1/1/91'	
	job_lvl	employee	<input type="checkbox"/>	= 100	

These grid entries produce the following WHERE clause in the statement in the SQL pane:

```
WHERE (hire_date < '01/01/91') AND  
      (job_lvl = 100)
```

4. In the **Or ...** grid column, enter conditions that you want to link with OR. For example, to add a condition that searches for another value in the job_lvl column, enter an additional value as shown here:

	Column	Table	Output	Criteria	Or...
	hire_date	employee	<input type="checkbox"/>	< '1/1/91'	
	job_lvl	employee	<input type="checkbox"/>	= 100	= 200

Adding a value in the **Or ...** column adds another condition to the WHERE clause in the statement in the **SQL** pane:

```
WHERE (hire_date < '01/01/91' ) AND
```

```
(job_lvl = 100) OR  
(job_lvl = 200)
```

[Specify conditions for groups](#)

You can limit the groups that appear in a query by specifying a condition that applies to groups as a whole— a HAVING clause. After the data has been grouped and aggregated, the conditions in the HAVING clause are applied. Only the groups that meet the conditions appear in the query.

For example, you might want to see the average price of all books for each publisher in a titles table, but only if the average price exceeds \$10.00. In that case, you could specify a HAVING clause with a condition such as `AVG(price) > 10`.

Note In some instances, you might want to exclude individual rows from groups before applying a condition to groups as a whole.

You can create complex conditions for a HAVING clause by using AND and OR to link conditions.

1. Specify the groups for your query.
2. If it is not already in the Grid pane, add the column on which you want to base the condition. (Most often the condition involves a column that is already a group or summary column.) You cannot use a column that is not part of an aggregate function or of the GROUP BY clause.
3. In the **Criteria** column, specify the condition to apply to the group.

The Query Designer automatically creates a HAVING clause in the statement in the SQL pane, such as in the following example:

```
SELECT pub_id, AVG(price)  
FROM titles  
GROUP BY pub_id  
HAVING (AVG(price) > 10)
```

4. Repeat steps 2 and 3 for each additional condition you want to specify.

[Show All](#)

Examples of using wildcard characters (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

You can search for patterns within data columns or expressions by using wildcard characters. For example, you can search for all employees whose last names begin with "Mac" or end with "son."

[Wildcard Characters](#)

You can use wildcard characters to search any columns that can be treated as text strings. Columns with the data type character can always be treated as text strings; those with the datetime data type can be treated as text strings.

To search for patterns, use the LIKE operator, and then substitute wildcard characters for one or more characters in the search string. You can use either of the following wildcard characters:

Wildcard character	Meaning
% (percent symbol)	Zero or more characters in that position
_ (underscore)	One character at that position

For example, to search for all names beginning with "Mac," you could specify the search condition `LIKE 'Mac%'`. To find names such as "Jan," "Jen," and "Jon," you could specify the search condition `LIKE 'J_n'`.

[Searching for Characters Used as Wildcard Characters](#)

In some cases, you might need to search for a string that contains one of the characters used as a wildcard character. For example, in a titles table you might want to find all the publications that contain the string "10%" as part of the title. Because "%" is part of the string you are searching for, you must specify that you mean it as a literal string and not a wildcard character.

To search for characters that can be interpreted as wildcard characters, you can specify an escape character. Place the escape character immediately in front of the "%" or "_" character that you mean literally. To specify the escape character, include an ESCAPE clause immediately after the LIKE search criterion. You can do this in either the Criteria column of the Grid pane or in the SQL pane.

For example, imagine that you want to find all titles that contain the string "10%". Suppose you want to define the character "#" as an escape character, which allows you to include "#" in front of the "%" character that is meant literally. You can enter this in the Grid pane:

```
LIKE '%10#%' ESCAPE '#'
```

The resulting WHERE clause in the SQL statement looks like this:

```
WHERE title LIKE '%10#%' ESCAPE '#'
```

Note You can define an escape character only in the SQL pane.



[Searching Datetime Columns](#)

When working with a datetime data type column, you can search any portion of the date or time, including text abbreviations of the month and complete years.

For example, you can use the following LIKE clause to search for all rows in which the date falls within 1994:

```
LIKE '%1994%'
```

The following searches for all rows in which the date falls within the month of January, regardless of year:

```
LIKE 'Jan%'
```



[Examples of Wildcard Searches](#)

The following examples illustrate the use of wildcard characters.

Search expression	Description
LIKE 'Mac%'	Finds values beginning with "Mac"
LIKE 'J%n'	Finds values starting with "J" and ending with "n"
LIKE '%son'	Finds values ending with "son"
LIKE '%sam%'	Finds values with "sam" anywhere in the string
LIKE '%Mar%'	Finds values in a datetime column that fall in the month of March, regardless of year
LIKE '%1994%'	Finds values in a datetime column for the year 1994.
	Finds

LIKE 'Mac_'

values with
exactly four
characters,
the first
three being
"Mac"

LIKE '_dam'

Finds
values with
exactly four
characters,
the last
three being
"dam"

LIKE '%s_n'

Finds
values
containing
"s" and "n"
at the end
of the
value, with
any one
character
between
them and
any number
of
characters
in front of
them

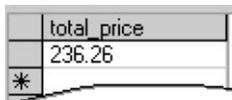
[Show All](#)

About aggregate functions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Overview of aggregate functions](#)

To summarize all the data in a table, you create an aggregate query that involves a function such as SUM() or AVG(). When you run the query, the result set contains a single row with the summary information. For example, you can calculate the total price of all books in the titles table by creating a query that sums the contents of the price column. The resulting query output might look like this:



	total_price
*	236.26

The corresponding SQL statement might look like this:

```
SELECT SUM(price) total_price  
FROM titles
```

When you use an aggregate function, by default the summary information includes all specified rows. In some instances, a result set includes non-unique rows. You can filter out non-unique rows by using the DISTINCT option of an aggregate function.

You can combine aggregate functions with other expressions to calculate other summary values.

For more information on aggregate functions, see the Microsoft SQL Server documentation.

[Summary of aggregate functions](#)

You can use the following aggregate functions:

Aggregate function	Description
AVG(<i>expression</i>)	Average of the values in a column. The column can contain only numeric data. Null values are ignored.
BINARY_CHECKSUM(<i>expression list</i>) BINARY_CHECKSUM(*)	The binary checksum value computed over a row of a table or over a list of expressions. BINARY CHECKSUM is used to detect changes to a row of a table.
CHECKSUM(<i>expression</i>)	The checksum value computed over a row of a table, or over a list of expressions. CHECKSUM is used for building hash indices.
CHECKSUM_AGG(<i>expression</i>)	The checksum value of a group. Null values are ignored.
COUNT(<i>expression</i>) COUNT(*) COUNT_BIG(<i>expression</i>), COUNT_BIG(*)	A count of the values in a column (if you specify a column name as <i>expr</i>) or of all rows in a table or group (if you specify *). COUNT(<i>expr</i>) and COUNT_BIG(<i>expr</i>) ignore null values, but COUNT(*) and COUNT_BIG(*) includes them in the count. COUNT always returns an int data type; COUNT_BIG always returns a bigint data type.
GROUPING(<i>column_name</i>)	An aggregate function that causes an additional column to be output with a value of 1 when the row is added by either the CUBE or ROLLUP operator, or 0 when the row is not the result of CUBE or ROLLUP.
MAX(<i>expression</i>)	Highest value in a column (last value alphabetically for text data types). Ignores null values.
MIN(<i>expression</i>)	Lowest value in a column (first value alphabetically for text data types). Ignores null values.
STDEV(<i>expression</i>) STDEVP(<i>expression</i>)	The statistical standard deviation of all values in the given expression. STDEVP returns the statistical standard deviation of

SUM(*expression*)

VAR(*expression*)

VARP(*expression*)

the population of all values.

Total of values in a column. The column can contain only numeric data.

The statistical variance of all values in the given expression. VARP returns the statistical variance of the population of all values.

[Show All](#)

About expressions in queries (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Overview of expressions](#)

You can use an expression anywhere in a query where you can use a column name. Expressions can calculate values to display, be part of search conditions, or combine the contents of data columns. An expression can consist of a mathematical calculation or a string, and can involve any combination of column names, literals, operators, or functions.

Examples of the use of expressions in a query include:

- In a products table, displaying a discounted price that is calculated by taking 10% off the retail price.
- Displaying only the first three digits— the area code— of a phone number.
- Displaying employee names in the format *last_name, first_name*.
- Joining two tables, an orders table and a products table, then sorting the query on the total price (order quantity times product price).
- In an orders table, copying and then deleting all orders that were shipped more than one year ago.

 [Expression structure and expression examples](#)

 [General guidelines for creating expressions](#)

An expression consists of any combination of column names, literals, operators, or functions. Follow these guidelines in combining elements to form expressions:

- Reference columns by typing their names. If your query uses more than one table and if you use a column name that is not unique, you must add the table name and a period to the column name. The following example shows the column name `job_id` qualified with the table name `employee`:

employee.job_id

- Include literal text by enclosing it in single quotation marks; no quotation marks are necessary for numbers.

Note The Query Designer can accept terms in double quotation marks, but might interpret them differently than you expect. In a Microsoft Access Project, the Query Designer always interprets double quotation marks as database object delimiters.

- Use standard arithmetic operators for numbers and a concatenation operator for combining strings.
- Include parentheses to establish precedence of operators.
- If you include a function, use these same guidelines for the arguments passed to the function. That is, reference columns by typing their names, enclose literal text in single quotation marks, and so on.
- If you pass column names as function arguments, be sure the data type of the column is appropriate for the function argument.
- You can include user-defined functions returning a scalar value in an expression.

[Expression examples](#)

The following table illustrates the use of expressions in a query.

Expression	Result
SELECT (price * .9) FROM products	Displays a discounted price (10% off the value in the price column).
SELECT (lname + , ' ' + fname) FROM employee	Displays the concatenated values of the last name and first name columns with a comma between them.
SELECT sales.qty, titles.price FROM sales INNER JOIN titles ON sales.title_id = titles.title_id ORDER BY (sales.qty *	After joining two tables, sorts the result set by the total value of an order (quantity times price).

titles.price)

```
SELECT au_lname,  
       au_fname  
FROM authors      Displays authors whose area code is in the San Francisco  
WHERE            area.  
       (SUBSTRING(phone,  
1, 3) = '415')  
SELECT ord_num,  
       ord_date  
FROM sales        Finds all orders in the sales table that were made in the  
WHERE            last 10 days. Today's date is returned by the GETDATE( )  
       (ord_date >= function.  
DATEADD(day, -10, GETDATE()))
```

[Operators for expressions](#)

You can use a variety of operators in constructing expressions for your query, including mathematical and text operators.

[Mathematical operators](#)

The following table lists the mathematical operators you can use in constructing an expression.

Operator	Meaning
+, -	Unary positive, negative
+	Addition
-	Subtraction
*	Multiplication
/	Division

Note You can use additional operators available, such as the "%" operator to determine modulo or remainders in arithmetic operations. For more information about the modulo arithmetic operator, see the Microsoft SQL Server documentation.

If you use more than one mathematical operator in an expression, the Query

Designer processes the expression according to the following operator precedence. To override the default precedence, use parentheses around the portion of the expression that is to be evaluated first. If more than one operator of the same level is included, the operators are evaluated left to right.

1. Unary + and -
2. * and /
3. + and -

[Text operator](#)

You can perform one operation on text: concatenation or the linking together of strings. You can concatenate strings and perform other operations (such as removing extraneous spaces) using a single operator. To concatenate a string, you can use the "+" operator in the Grid pane.

[Predefined variables for expressions](#)

In addition to using column names, literals, operators, and functions in an expression, you can use predefined variables that have defined meanings or values. For example, you can use a predefined variable to display the user name for the current user or to search for data columns that contain no value (null).

The list below includes examples of predefined variables.

Predefined Variable	Description	Example
	The user name of the current user	
CURRENT_USER	<p>Security Use this feature with caution. Sensitive or confidential information could be revealed to</p> <p>UPDATE accounts SET salesperson = CURRENT_USER WHERE region = 'NW'</p> <p>Changes the salesperson column for all acc region to the name of the current user.</p>	

other users.

NULL

A null value, used in the search

```
SELECT emp_id, lname, fname, minit
FROM employee
WHERE minit IS NULL
```

expressions

IS NULL and Finds authors who have no middle initial.

IS NOT

NULL

Note The keyword IS is a special operator fo

[Show All](#)

Group rows in query results (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

If you want to create subtotals or show other summary information for subsets of a table, you create groups using an aggregate query. Each group summarizes the data for all the rows in the table that have the same value.

For example, you might want to see the average price of a book in the `titles` table, but break the results down by publisher. To do so, you group the query by publisher (for example, `pub_id`). The resulting query output might look like this:

pub_id	Avg Price
0736	9.784
0877	15.41
1389	18.976

When you group data, you can display only summary or grouped data, such as:

- The values of the grouped columns (those that appear in the `GROUP BY` clause). In the example above, `pub_id` is the grouped column.
- Values produced by aggregate functions such as `SUM()` and `AVG()`. In the example above, the second column is produced by using the `AVG()` function with the `price` column.

You cannot display values from individual rows. For example, if you group only by publisher, you cannot also display individual titles in the query. Therefore, if you add columns to the query output, the Query Designer automatically adds them to the `GROUP BY` clause of the statement in the SQL pane. If you want a column to be aggregated instead, you can specify an aggregate function for that column.

If you group by more than one column, each group in the query shows the aggregate values for all grouping columns.

For example, the following query against the titles table groups by publisher (pub_id) and also by book type (type). The query results are ordered by publisher and show summary information for each different type of book that the publisher produces:

```
SELECT pub_id, type, SUM(price) Total_price
FROM titles
GROUP BY pub_id, type
```

The resulting output might look like this:

pub_id	type	Total_price
0736	business	2.99
0736	psychology	45.93
0877	mod_cook	22.98
0877	psychology	21.59
0877	trad_cook	47.89
1389	business	51.93
1389	popular_cook	

[Group rows](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Start the query by adding the tables, views, or functions you want to summarize to the Diagram pane.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables**, **Views**, or **Functions**

- tab.
4. In the list of items, double-click each item you want to add.
 5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view, or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL** to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the Grid pane.

4. Add the column or columns you want to group to the **Grid** pane. If you want the column to appear in the query output, be sure that the **Output** column is selected for output.

The Query Designer adds a GROUP BY clause to the statement in the SQL pane. For example, the SQL statement might look like this:

```
SELECT pub_id
FROM titles
GROUP BY pub_id
```

5. Add the column or columns you want to aggregate to the **Grid** pane. Be sure that the column is marked for output.
6. In the **Group By** grid cell for the column that is going to be aggregated, select the appropriate aggregate function.

The Query Designer automatically assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one.

Column	Alias	Table	Output	Group By
pub_id		titles (dbo)	<input checked="" type="checkbox"/>	Group By
price	Total price	titles (dbo)	<input checked="" type="checkbox"/>	Sum
			<input type="checkbox"/>	

The corresponding statement in the SQL pane might look like this:

```
SELECT pub_id, SUM(price) AS Totalprice
FROM titles
GROUP BY pub_id
```

[Show All](#)

Summarize or aggregate values for all rows in a query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

Using an aggregate function, you can create a summary for all the values in a query. For example, you can create a query such as the following to display the total price for all books in the `titles` table:

```
SELECT SUM(price)
FROM titles
```

You can create multiple aggregations in the same query by using aggregate functions with more than one column. For example, you can create a query that calculates the total of the price column and the average of the discount column.

You can also aggregate the same column in different ways (such as totaling, counting, and averaging) in the same query. For example, the following query averages and summarizes the price column from the `titles` table:

```
SELECT AVG(price), SUM(price)
FROM titles
```

If you add a search condition, you can aggregate the subset of rows that meet that condition.

Note You can also count all the rows in the table or the ones that meet a specific condition.

When you create a single aggregation value for all rows in a query, you display only the aggregate values themselves. For example, if you are totaling the value

of the price column of the titles table, you would not also display individual titles, publisher names, and so on.

Note If you are subtotaling— that is, creating groups— you can display column values for each group.

[Aggregate values for all rows](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Be sure the tables, views, or functions you want to aggregate are already present in the Diagram pane.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the Grid pane.
4. Add the column you want to aggregate to the Grid pane. Be sure that the column is marked for output. The Query Designer automatically assigns a column alias to the column you are summarizing. You can replace this alias with a more meaningful one.
5. In the **Group By** grid column, select the appropriate aggregate function, such as: **Sum**, **Avg**, **Min**, **Max**, **Count**. If you want to aggregate only unique rows in the result set, choose an aggregate function with the **DISTINCT** options, such as **Min Distinct**. Do not choose **Group By**, **Expression**, or **Where**, because those options do not apply when you are aggregating all rows.

The Query Designer replaces the column name in the statement in the SQL pane with the aggregate function that you specify. For example, the SQL statement might look like this:

```
SELECT SUM(price)
FROM titles
```

6. If you want to create more than one aggregation in the query, repeat steps 3 and 4.

When you add another column to the query output list or order by list, the Query Designer automatically fills the term **Group By** into the **Group By** column of the grid. Select the appropriate aggregate function.

7. Add search conditions, if any, to specify the subset of rows you want to summarize.

When you execute the query, the datasheet displays the aggregations that you specified.

Note The Query Designer maintains aggregate functions as part of the SQL statement in the SQL pane until you explicitly turn off Group By mode. Therefore, if you modify your query by changing its type or by changing which tables, views, or functions are present in the diagram pane, the resulting query might include invalid aggregate functions.

[Show All](#)

Summarize or aggregate values using custom expressions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

In addition to using aggregate functions to aggregate data, you can create custom expressions to produce aggregate values. You can use custom expressions in place of aggregate functions anywhere in an aggregate query.

For example, in the `titles` table you might want to create a query that shows not just the average price, but what the average price would be if it were discounted.

You cannot include an expression that is based on calculations involving only individual rows in the table; the expression must be based on an aggregate value, because only the aggregate values are available at the time the expression is calculated.

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Specify the groups for your query.

[How?](#)

[Background information](#)

If you want to create subtotals or show other summary information for subsets of a table, you create groups using an aggregate query. Each group summarizes the data for all the rows in the table that have the same value.

For example, you might want to see the average price of a book in the `titles` table, but break the results down by publisher. To do so, you group the query by publisher (for example, `pub_id`). The resulting query output might look like this:

pub_id	Avg Price
0736	9.784
0877	15.41
1389	18.976

When you group data, you can display only summary or grouped data, such as:

- The values of the grouped columns (those that appear in the GROUP BY clause). In the example above, pub_id is the grouped column.
- Values produced by aggregate functions such as SUM() and AVG(). In the example above, the second column is produced by using the AVG() function with the price column.

You cannot display values from individual rows. For example, if you group only by publisher, you cannot also display individual titles in the query. Therefore, if you add columns to the query output, the Query Designer automatically adds them to the GROUP BY clause of the statement in the SQL pane. If you want a column to be aggregated instead, you can specify an aggregate function for that column.

If you group by more than one column, each group in the query shows the aggregate values for all grouping columns.

For example, the following query against the titles table groups by publisher (pub_id) and also by book type (type). The query results are ordered by publisher and show summary information for each different type of book that the publisher produces:

```
SELECT pub_id, type, SUM(price) Total_price
FROM titles
GROUP BY pub_id, type
```

The resulting output might look like this:

pub_id	type	Total_price
0736	business	2.99
0736	psychology	45.93
0877	mod_cook	22.98
0877	psychology	21.59
0877	trad_cook	47.89
1389	business	51.93
1389	popular_cook	

[Group rows](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Start the query by adding the tables, views, or functions you want to summarize to the Diagram pane.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables, Views, or Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, Grid pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view,

or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL** to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the Grid pane.
 4. Add the column or columns you want to group to the **Grid** pane. If you want the column to appear in the query output, be sure that the **Output** column is selected for output.

The Query Designer adds a GROUP BY clause to the statement in the SQL pane. For example, the SQL statement might look like this:

```
SELECT pub_id
FROM titles
GROUP BY pub_id
```

5. Add the column or columns you want to aggregate to the **Grid** pane. Be sure that the column is marked for output.
6. In the **Group By** grid cell for the column that is going to be aggregated, select the appropriate aggregate function.

The Query Designer automatically assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one.

Column	Alias	Table	Output	Group By
pub_id		titles (dbo)	<input checked="" type="checkbox"/>	Group By
price	Total price	titles (dbo)	<input checked="" type="checkbox"/>	Sum
			<input type="checkbox"/>	

The corresponding statement in the SQL pane might look like this:

```
SELECT pub_id, SUM(price) AS Totalprice
FROM titles
GROUP BY pub_id
```

3. Move to a blank row of the Grid Pane, and then type the expression in the **Columns** column.

The Query Designer automatically assigns a column alias to the expression to create a useful column heading in query output.

4. In the **Group By** column for the expression, select **Expression**.

[Show All](#)

Use HAVING and WHERE clauses in the same query (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

In some instances, you might want to exclude individual rows from groups (using a WHERE clause) before applying a condition to groups as a whole (using a HAVING clause).

A HAVING clause is like a WHERE clause, but applies only to groups as a whole (that is, to the rows in the result set representing groups), whereas the WHERE clause applies to individual rows. A query can contain both a WHERE clause and a HAVING clause. In that case:

- The WHERE clause is applied first to the individual rows in the tables, views, or functions in the diagram pane. Only the rows that meet the conditions in the WHERE clause are grouped.
- The HAVING clause is then applied to the rows in the result set that are produced by grouping. Only the groups that meet the HAVING conditions appear in the query output. You can apply a HAVING clause only to columns that also appear in the GROUP BY clause or in an aggregate function.

For example, imagine that you are joining the `titles` and `publishers` tables to create a query showing the average book price for a set of publishers. You want to see the average price for only a specific set of publishers— perhaps only the publishers in the state of California. And even then, you want to see the average price only if it is over \$10.00.

You can establish the first condition by including a WHERE clause, which discards any publishers that are not in California, before calculating average prices. The second condition requires a HAVING clause, because the condition is based on the results of grouping and summarizing the data. The resulting SQL

statement might look like this:

```
SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
  ON titles.pub_id = publishers.pub_id
WHERE publishers.state = 'CA'
GROUP BY titles.pub_id
HAVING AVG(price) > 10
```

You can create both **HAVING** and **WHERE** clauses in the **Grid** pane of the Query Designer. By default, if you specify a search condition for a column, the condition becomes part of the **HAVING** clause. However, you can change the condition to be a **WHERE** clause.

You can create a **WHERE** clause and **HAVING** clause involving the same column. To do so, you must add the column twice to the **Grid** pane, then specify one instance as part of the **HAVING** clause and the other instance as part of the **WHERE** clause.

[Specify a WHERE condition in an aggregate query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Specify the groups for your query.

[How?](#)

[Background information](#)

If you want to create subtotals or show other summary information for subsets of a table, you create groups using an aggregate query. Each group summarizes the data for all the rows in the table that have the same value.

For example, you might want to see the average price of a book in the `titles` table, but break the results down by publisher. To do so, you group the query by publisher (for example, `pub_id`). The resulting query output might look like this:

pub_id	Avg Price
0736	9.784
0877	15.41
1389	18.976

When you group data, you can display only summary or grouped data, such as:

- The values of the grouped columns (those that appear in the GROUP BY clause). In the example above, pub_id is the grouped column.
- Values produced by aggregate functions such as SUM() and AVG(). In the example above, the second column is produced by using the AVG() function with the price column.

You cannot display values from individual rows. For example, if you group only by publisher, you cannot also display individual titles in the query. Therefore, if you add columns to the query output, the Query Designer automatically adds them to the GROUP BY clause of the statement in the SQL pane. If you want a column to be aggregated instead, you can specify an aggregate function for that column.

If you group by more than one column, each group in the query shows the aggregate values for all grouping columns.

For example, the following query against the titles table groups by publisher (pub_id) and also by book type (type). The query results are ordered by publisher and show summary information for each different type of book that the publisher produces:

```
SELECT pub_id, type, SUM(price) Total_price
FROM titles
GROUP BY pub_id, type
```

The resulting output might look like this:

pub_id	type	Total_price
0736	business	2.99
0736	psychology	45.93
0877	mod_cook	22.98
0877	psychology	21.59
0877	trad_cook	47.89
1389	business	51.93
1389	popular_cook	

[Group rows](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Start the query by adding the tables, views, or functions you want to summarize to the Diagram pane.

[How?](#)

When you create a query, you are retrieving data from a table, view, or function. To work with any of these objects in your query, you add them to the Diagram pane.

[Add a table, view, or user-defined function to the query](#)

1. In the Database window, click **Queries**  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. In the Diagram pane, right-click the background and choose **Add Table** from the shortcut menu.
3. In the **Add Table** dialog box, select the **Tables, Views, or Functions** tab.
4. In the list of items, double-click each item you want to add.
5. When you finish adding items, click **Close**.

The Query Designer updates the Diagram pane, **Grid** pane, and SQL pane accordingly.

Alternatively, you can drag objects onto the Diagram pane. You can drag a table, view, or inline function from the database window.

You can also drag columns or tables from the Database Designer or paste them from the Clipboard.

Tables and views are automatically added to the query when you reference them in the statement in the SQL pane.

The Query Designer will not display data columns for an table, view,

or inline function if you do not have sufficient access rights. In such cases, only a title bar and the * (All Columns) check box are displayed for the table, view, or inline function.

[Add an existing query to a new query](#)

1. If necessary, click **SQL** to show the **SQL** pane.
 2. In the SQL pane, type a right and left parentheses () after the word FROM.
 3. Open the Query designer for the existing query . (You now have two Query Designers open.)
 4. Display the SQL pane for the inner query – the existing query you are including in the new, outer query.
 5. Select all the text in the SQL pane, and copy it to the Clipboard.
 6. Click in the SQL pane of the new query, situate the cursor between the parentheses you added, and paste the contents of the Clipboard.
 7. Still in the SQL pane, add an alias after the right parenthesis. For more information on SQL aliases and subqueries, see the Microsoft SQL Server documentation.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the **Grid** pane.
 4. Add the column or columns you want to group to the **Grid** pane. If you want the column to appear in the query output, be sure that the **Output** column is selected for output.

The Query Designer adds a GROUP BY clause to the statement in the SQL pane. For example, the SQL statement might look like this:

```
SELECT pub_id
FROM titles
GROUP BY pub_id
```

5. Add the column or columns you want to aggregate to the **Grid** pane. Be sure that the column is marked for output.
6. In the **Group By** grid cell for the column that is going to be aggregated, select the appropriate aggregate function.

The Query Designer automatically assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one.

Column	Alias	Table	Output	Group By
pub_id		titles (dbo)	<input checked="" type="checkbox"/>	Group By
price	Total price	titles (dbo)	<input checked="" type="checkbox"/>	Sum
			<input type="checkbox"/>	

The corresponding statement in the SQL pane might look like this:

```
SELECT pub_id, SUM(price) AS Totalprice
FROM titles
GROUP BY pub_id
```

3. If it is not already in the **Grid** pane, add the column on which you want to base the WHERE condition.
4. Clear the **Output** column unless the data column is part of the GROUP BY clause or included in an aggregate function.
5. In the **Criteria** column, specify the WHERE condition. The Query Designer adds the condition to the HAVING clause of the SQL statement.

Note The query shown in the example for this procedure joins two tables, titles and publishers.

At this point in the query, the SQL statement contains a HAVING clause:

```
SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
ON titles.pub_id = publishers.pub_id
GROUP BY titles.pub_id
HAVING publishers.state = 'CA'
```

6. In the **Group By** column, select **Where** from the list of group and summary options. The Query Designer removes the condition from the HAVING clause in the SQL statement and adds it to the WHERE clause.

The SQL statement changes to include a WHERE clause instead:

```
SELECT titles.pub_id, AVG(titles.price)
FROM titles INNER JOIN publishers
ON titles.pub_id = publishers.pub_id
WHERE publishers.state = 'CA'
```

```
GROUP BY titles.pub_id
```

[Show All](#)

Transact SQL scalar functions for expressions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Overview of scalar functions](#)

You can call a number of scalar functions when you are building an expression, including:

- String (character) functions
- Date functions
- Mathematical functions
- System functions
- Other functions, such as those to convert data from one type to another

The Query Designer can help you work with functions by:

- Correctly inserting quotation marks in function arguments
- Validating the data types of arguments
- Validating the data types of return values

The following information is a list of commonly-used scalar functions. For a complete list of SQL Server scalar functions, see the Transact SQL reference *Functions* help topic in the Microsoft SQL Server documentation

Note You can also use a special set of functions, the aggregate functions such as SUM() and AVG(), to create queries that summarize data.

 [String functions](#)

Function	Description	Example
----------	-------------	---------

```
SELECT UPPER(substring(lname, 1, 1)) +  
       LOWER(substring (lname, 2, 99))
```

LCASE(), LOWER()	Converts strings to lowercase	FROM employee
		Displays a last name after the first character is converted to uppercase and the remaining characters to lowercase
LTRIM()	Removes leading spaces from a string	SELECT stor_name, LTRIM (stor_address) FROM stores
		Displays an address column after extraneous spaces from the front.
SUBSTRING())	Extracts one or more characters from a string	SELECT SUBSTRING (phone, 1, 3) FROM employee
		Displays the first three characters (the area code) of number.
UCASE(), UPPER()	Converts strings to uppercase	SELECT * FROM employee WHERE UPPER (lname) = 'SMITH'
		Converts the contents of the lname column to uppercase and compares them to a specific value (avoids mismatch search is case sensitive).

 [Date functions](#)

Function	Description	Example
DATEDIFF())	Calculates an interval between two dates.	SELECT fname, lname, hire_date FROM employee WHERE DATEDIFF (year, hire_date, getdate()) > 5
		Locates all employees hired more than five years ago

Returns the specified portion of a date or datetime column, including the day, month, or year.

```
SELECT DATEPART(year, hire_date)
FROM employee
```

DATEPART() Displays only the year in which an employee was hired (full date).

Returns the current date in datetime format. This function is useful as input for many other date functions, such as calculating an interval forward or backward from today.

```
SELECT order_id
FROM orders
WHERE order_date = GETDATE()
```

CURDATE(), **GETDATE()** or **DATE()** Displays orders placed today.

[Mathematical functions](#)

Note You can also use aggregate functions, such as **AVG()**, **COUNT()**, **MAX()**, **MIN()**, and **SUM()** to create averages and totals in your report.

Function	Description	Example
ROUND()	Rounds a number off to the specified number of	<pre>SELECT ROUND(qty * (price * discount), 2) FROM sales</pre> <p>Displays a total price based on a discount, then rounds t</p>

	decimal places	off to two decimal places.
	Rounds a number down to the nearest (smallest) whole number	UPDATE titles SET price = FLOOR (price)
FLOOR()		Rounds all prices in the titles table down to the nearest number.
	Rounds a number up to the nearest whole number	INSERT INTO archivetable SELECT title, CEILING (price) FROM titles
CEILING()		Copies the title and the price (rounded up to the nearest whole number) from the titles table to the archivetable table.

[System functions](#)

Function	Description	Example
DATALENGTH()	Returns the number of bytes used by the specified expression	<pre>SELECT DATALENGTH(au_lname + ', ' + au_fname) FROM authors</pre> <p>Lists the number of bytes required for the combination names.</p>
USER() , USER_NAME()	Returns the current user name	<pre>SELECT company_name, city, phone FROM customers WHERE salesperson = USER_NAME()</pre> <p>Use this feature with caution. Sensitive or</p>

confidential information could be revealed to other users. Creates a list of customers for the salesperson w

[Other functions](#)

Function	Description	Example
CONVERT ()	Converts data from one data type into another. Useful to format data or to use the contents of a data column as an argument in a function that requires a different data type.	<pre>SELECT 'Hired: ' + CONVERT(char (11), hire_da FROM employee</pre>
SOUNDEX ()	Returns the Soundex code for the specified expression, which you can use to create "sounds like"	<pre>SELECT au_lname, au_fname FROM authors WHERE SOUNDEX(au_fname) = 'M240'</pre> Searches for names that sound like "Michael".

searches.

Converts

numeric
data into a
character

```
SELECT str(job_id) + ' ' +  
       str(job_lvl)  
FROM employee
```

STR()

string so

you can
manipulate
it with text
operators.

Displays the job_id and job_lvl columns (both numeric) as a single string.

[Show All](#)

Count rows in a query (ADP)

You can count rows in a query to determine:

- The total number of rows, for example, a count of all the books in a `titles` query.
- The number of rows that meet a specific condition, for example, the number of books by one publisher in a `titles` query.
- The number of values in a particular column. When you count values in a column, nulls are not included in the count.

[Count all rows](#)

1. In the Database window, click **Queries**  under **Objects**, and then click **New** on the database window toolbar.
2. Add the table, view, or function you want to summarize in the Diagram pane.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the Grid pane.
4. Select *** (All Columns)** in the rectangle representing the table, view, or function.

The Query Designer automatically fills the term **Count** into the **Group By** column in the Grid pane and assigns a column alias to the column you are summarizing. You can replace this automatically generated alias with a more meaningful one.

[Count all the rows that meet a condition](#)

1. In the Database window, click **Queries**  under **Objects**, and then click **New** on the database window toolbar.
2. Add the table, view, or function you want to summarize in the Diagram pane.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to

the grid in the Grid pane.

4. Select * (**All Columns**) in the rectangle representing the table, view, or function.

The Query Designer automatically fills the term **Count** into the **Group By** column in the **Grid** pane and assigns a column alias to the column you are summarizing.

5. Add the data column that you want to search, and then clear the check box in the **Output** column.

The Query Designer automatically fills the term **Group By** into the **Group By** column of the grid.

6. Change **Group By** in the **Group By** column to **Where**.
7. In the **Criteria** column for the data column to search, enter the search condition.

[Count the values in a column](#)

1. In the Database window, click **Queries**  under **Objects**, and then click **New** on the database window toolbar.
2. Add the table, view, or function you want to summarize in the Diagram pane.
3. Right-click the background of the Diagram pane, then choose **Group By** from the shortcut menu. The Query Designer adds a **Group By** column to the grid in the Grid pane.
4. Add the column that you want to count to the **Grid** pane.

The Query Designer automatically fills the term **Group By** into the **Group By** column of the grid.

5. Change **Group By** in the **Group By** column to **Count**.

Note To count only unique values, choose **Count Distinct**.

[Show All](#)

About handling concurrent updates (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

In a multiuser environment, more than one person might be working with the same record at the same time. Because other people can change or even delete the same data that you're trying to edit, you might occasionally encounter data conflicts with others as they work. A [Microsoft Access project](#), by default, uses a technique called optimistic record locking to handle record contention.

If you modify a record, and someone updates that record before you save it, Access displays a message box and warns you that if you save the record, you will overwrite the changes that the other user has made. You can do one of the following:

- Press **Save Record** to save the record and overwrite the other user's changes.
- Press **Copy to Clipboard** to first examine what the other user has entered as changes so that you can reconcile them with your current edits.
- Press **Drop Changes** to ignore your changes and accept the other user's changes.

Because other users may add, modify, or delete records that you are accessing, you should periodically refresh the display of data in Datasheet or Form view by using the **Refresh** command (**Records** menu) to re-query and see up-to-date data.

[Show All](#)

Set the number of records retrieved in an Access project (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

To improve performance, you can set the maximum number of records downloaded from a SQL Server database when working with data in a form or datasheet in a [Microsoft Access project](#).

1. Open a datasheet or form in browse mode.
2. Do one of the following:
 - Select **Maximum Records** from the **Records** menu.
 - Click **maximum record limit**  to the right of the [navigation buttons](#).
3. In the **Set Maximum Record Count** dialog box, type in the maximum number of records in the text box. Typing 0 means there is no Limit and all records are retrieved.

Note Be careful when removing the limit— it could decrease the performance of your Access project— especially if the [recordset](#) of the form or datasheet is large. However, if you are exporting data to another Access project, or running an append or make-table query to copy data, you may want to set the maximum record limit to 0 if you need to retrieve all records.

 [Tip](#)

If you set the maximum record limit to a high value, and browsing your data takes a long time, you can cancel the operation by clicking **Cancel Query**  or pressing CTRL+BREAK.

Notes

- If you are using a form, Microsoft Access saves the setting with the form if you adjust the setting. The default maximum record limit is 10,000 for all new forms and datasheets. To change the default limitation the **Tools** menu,

click **Options**, click the **Advanced** Tab, and change the default number in the **Default max records** text box. Entering a value of 0 means that there is no limit by default. You can also set the default value for an individual form by using the **MaxRecords** property.

- Access displays the maximum record limit button by default, but you can hide this button by setting the **MaxRecButton** property of a form or datasheet to False. This property has no effect on the menu command.
- Access loads records from the SQL Server database asynchronously, which means you can interact with Access before all the records are retrieved. Access displays the number of records it is loading from the SQL Server database next to the maximum record limit button.

[Show All](#)

About copying data and databases in an Access project (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

There are four ways to copy data and database information in a Microsoft Access project:

- Import one or more tables.
- Export one or more tables.
- Copy a database.
- Transfer a database.

[Importing and exporting data](#)

When you import or export data (or copy and paste), the following information gets copied along with the data:

- The table name. However, if the table name already exists, Access adds a numeric suffix (for example, Customers becomes Customers1).
- The primary key of the table. If the primary key name already exists, Access tries to recreate the primary key based on the table name by adding a PK_ prefix to the primary key name (for example, PK_Customers). If that name already exists, then Access adds a numeric suffix to the name (for example, PK_Customers1).

Note Primary keys are not imported or exported if your Access project is connected to a SQL server version 6.5 database.

- Column name, data type, length, precision, scale, and [Null](#) value settings. User-defined data types are not imported or exported, but rather are converted to their base types.

When you import or export data (or copy and paste), the following information does not get copied along with the data:

- Extended properties, such as default values, validation text constraints, and lookup values.

[Copy or transfer a database](#)

You can copy or transfer an SQL Server version 7.0 or later database. Both these operations not only copy the data and data definitions, but, unlike import or export operations, also copy extended properties such as default values, text constraints, and lookup values.

There are different requirements for copy and transfer database operations you need to be aware of before deciding which one best fits your needs. Copying a database is generally faster than transferring a database, but copying a database has more requirements.

[Requirements for copying a database](#)

- You must disconnect all applications and users before you copy the database file.
- All windows and views except the [database window](#) must be closed.
- The current database must not be replicated.
- The source server database must be Microsoft SQL Server version 7.0 or later, or SQL Server 2000 Desktop Engine, and must be a local computer.
- The SQL Server database on the source server must be a single file database.
- You must be a member of the sysadmin role on both the source and destination SQL servers.

[Requirements for transferring a database](#)

- You must be a member of the sysadmin role on the destination server (no special role is required on the source server).
- The current SQL server connected to the Access project and the destination server you are transferring the database to must be SQL Server version 7.0 or later.

Note Linked Servers are not transferred during a database transfer operation.

 [About the SQL Distributed Management Objects \(SQL-DMO\) version 8.0 requirement](#)

You must have SQL Distributed Management Objects (SQL-DMO) version 8.0 installed on your local computer to import, export, copy or transfer data and databases. SQL-DMO version 8.0 is a component of SQL Server 2000, but is not automatically installed on your computer when you install Microsoft Office 2003. This situation can occur if you have not installed Microsoft SQL Server 2000 or SQL Server 2000 Desktop Engine on your local computer. To resolve this, install Microsoft SQL Server 2000 or SQL Server 2000 Desktop Engine on your local computer. You can install the SQL Server 2000 Desktop Engine from the \MSDE2000 folder on the Microsoft Office 2003 CD-ROM.

[Show All](#)

Copy or transfer a SQL Server database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Copy a SQL Server database](#)

When you copy a Microsoft SQL Server database 7.0 or later database, Access detaches the SQL Server database file (.mdf) connected to your Access project (.adp), copies it to a new database, and then re-attaches it. For more information on detaching and re-attaching a database, see the SQL Server documentation.

[Before you copy](#)

To copy a SQL Server database:

- You must have Microsoft SQL Server version 7.0 or later installed on your local computer.
- All users must be disconnected from the database.
- The database must be a single file database.
- The database must not be replicated.
- You must be a member of the sysadmin role on both the source and destination SQL server.
- You must have adequate disk space to create the SQL Server database on the destination server.

1. Close all windows and views except the [Database window](#).
2. Point to the **Database Utilities** command on the **Tools** menu and click **Copy Database File**.

Access displays the **Open** dialog box.

3. Enter a master data file name. The default file name is "Copy <database name>.mdf" and the default location is the default database folder.

4. Press **OK**.

If other users are connected to the current SQL database (whether or not they are using the Access project), Access prompts you to disconnect all active connects th the database.

Caution Be careful disconnecting other users from a database. Disconnecting a user unexpectedly could lead to data loss.

Access detaches the database with the skipchecks option set to 'skipchecks' and copies the database with the new name. Whether or not, the copy operation succeeds or fails (for example, because of a sharing violation), Access re-attaches the current SQL Server database file and reconnects the Access project to the database.

[Transfer a SQL Server database](#)

You can transfer a SQL Server Database version 7.0 or later database file (.mdf) connected to an Access project (.adp). For more information on transferring a database, see the SQL Server documentation.

Note To transfer a database, you must have Microsoft SQL Server version 7.0 or later installed on you local compute, and be a member of the sysadmin role on the destination server.

1. Close all open windows except the [database window](#).
2. Point to the **Database Utilities** command on the **Tools** menu and click **Transfer Database**.
3. Follow the instructions in the Transfer Database Wizard:

What SQL Server would you like to use for this database? Type the name of the server you would like to use.

Use Trusted Connection You can use a trusted connection, that is, SQL Server can integrate with the Windows 2000 or later operating system security to provide a single log on to the network and the database.

Login ID and Password If you don't use a trusted connection, type the login ID and password of an account with CREATE DATABASE privileges on the server.

What do you want to name your new SQL Server database? Type the name of the new SQL Server database. Access revises the name if it conflicts with an existing database name and adds a numbered suffix (mydatabase 1, for example).

Include Data Clear **Include Data** if you want to just transfer data definitions and extended properties (In this case, database diagrams are not transferred).

Access displays the **Transfer Status** dialog box. If you cancel the transfer operation before it is complete, the copy of the database may be in an unknown state. Access prompts you to either keep the database or drop it.

[Show All](#)

Troubleshoot copying, transferring, and linking to a SQL Server database (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Copying and transferring a SQL Server database

[I'm having problems copying a SQL database.](#)

Possible reasons are:

- Other users are connected to the [Microsoft Access project](#).

Disconnect all applications and users before you copy the database file. Be careful doing this. Make sure you warn users before disconnecting them to prevent possible data loss.

- One or more database objects are currently open.

Close all windows and views except the [database window](#).

- The current database is replicated. You cannot copy a replicated SQL Server database.

You can use Microsoft SQL Server Enterprise Manager to remove publications and subscriptions before copying the database file.

- Copying an SQL Server database requires Microsoft SQL Server version 7.0 or SQL Server 2000 Desktop Engine on the local computer. To resolve this, you can do the following:
 - If you want to make a copy of a database on a remote server, point to **Database Utilities** on the **Tools** menu, and then click **Transfer Database**.
 - You can install Microsoft SQL Server 2000 Desktop Engine on your local computer from the \MSDE2000 folder on the Microsoft Office 2003 CD-ROM.
 - If you are connected to a SQL Server 6.5 database, you can use SQL Server Enterprise Manager to detach the database, copy it to the new server, and then reattach it.
- The SQL Server database has more than one data file associated with the current database. You can only copy a single file database.

- You are not a member of the sysadmin role. You must be a member of the sysadmin role on both the source and destination server to copy a SQL Server database.
- A database with the same name already exists on the destination server. Provide a new name for the database.
- You don't have enough disk space. Access cancels a copy database operation if you don't have enough disk space to create the new database.

Make sure you have adequate disk space on the destination server before doing a copy database operation.

If you encounter errors during the database copy operation, open the transfer log files created in the same folder as the Access project for more details.

[I'm having problems transferring a SQL database.](#)

Possible reasons are:

- Either the source or destination database is SQL Server version 6.5. The current SQL server connected to the Access project and the destination server you are transferring the database to must be SQL Server version 7.0 or later.
- You must be a member of the sysadmin role on the destination server. Although being a member of the sysadmin role is not required on the source server, you must be a member of the sysadmin role on the destination server to transfer a SQL Server database.
- A database with the same name already exists on the destination server. Provide a new name for the database.
- You don't have enough disk space. Access cancels a transfer database operation if you don't have enough disk space to create the new database.

Make sure you have adequate disk space on the destination server before doing a transfer database operation.

If you encounter errors during the database transfer operation, open the transfer log files created in the same folder as the Access project for more details.

[My extended properties were not copied when I transferred my SQL](#)

[Server database.](#)

To copy extended properties, Access requires Microsoft Data Access Components (MDAC) 2.6 on your local computer. If you don't have SQL Server 2000 or SQL Server 2000 Desktop Engine installed on your local computer, you may only have MDAC 2.5.

To install MDAC 2.6, you can install SQL Server 2000 Desktop Engine on your local computer from the \MSDE2000 folder on the Microsoft Office 2003 CD-ROM.

Linking to a SQL Server database

[I'm having problems linking to text files using the Linked Table Wizard.](#)

You cannot link [fixed-width text files](#) to SQL Server using the Linked Table Wizard. If you can, convert the text file to a [delimited text file](#). You might consider importing the fixed width text file on a regular basis instead. Importing the text file will also improve performance, especially if the table you create has indexes.

The Link Table Wizard always uses the comma (,) character as a list separator with delimited text files. You may have changed the list separator on your machine by changing the **Regional Settings** through **Windows Control Panel**. For example, the German list separator is a semicolon (;) character. If you can, convert the list separator to a comma (,) character before linking.

[When I link to a spreadsheet, text file, or HTML file, performance is very slow.](#)

You may have selected **Linked SQL** in the first screen of the Link Table Wizard, which creates a linked server. If you select **Transact SQL**, Access creates an OLE DB data source that uses Transact SQL functions to create an ad hoc read-only connection to the external data source. Access uses either the OPENDATASOURCE (SQL Server 2000) or OPENROWSET (SQL Server 7.0) Transact SQL functions. If you are linking to a spreadsheet, text file, or HTML file, consider using this option because the performance is generally better than creating a linked server. You might consider importing the data source on a regular basis instead. Importing the data source will also improve performance, especially if the table you create has indexes.

[I'm getting the message "You cannot link to file '<filename>'."](#)

Your Access project is connected to a SQL Server database on a remote computer. You can only link to tables when your Access project is connected to a SQL Server version 7.0 or later database or SQL Server 2000 Desktop Engine running on your local computer.

[When I link to a spreadsheet, text file, or HTML file my first row of data is missing.](#)

When you link to a spreadsheet, text file, or HTML file, Access assumes that the first row contains field names, not data, and uses the first row of data to make the column headers. Make sure the spreadsheet, text file, or HTML file has column headers, and then re-link the data source.

[My extended properties are not being recognized or saved in my linked table.](#)

Extended properties are ignored in a linked table. Consider transferring or copying the SQL Server database.

[Show All](#)

Troubleshoot updating data (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[I get the message, "Error validating constraint."](#)

Possible reasons are:

- The constraint expression is blank.
- The text of the **Constraint Expression** property does not conform to the constraint syntax recognized by SQL Server. Some common reasons are:
 - The expression does not resolve to a value of TRUE or FALSE.
 - The expression misspells a column name, or refers to a column in another table.
 - The expression misspells the name of a function or database constant.

You must enter a constraint in the **Constraints Expression** dialog box before continuing to define additional properties for your table. For more information on the syntax of constraint expressions, see the SQL Server documentation.

[I'm having problems with the default data type.](#)

- The default data type you selected in the **Tables/Queries** tab of the **Options** command under the **Tools** menu may not be supported in all versions of Microsoft SQL Server. For example, The **bigint** and **sql_variant** data types are only supported in SQL Server 2000. If an earlier version of SQL server doesn't support the data type you have chosen, Access replaces it as the following table shows.

SQL Server 2000 SQL Server 7.0 SQL Server 6.5

bigint	int	int
nchar	nchar	varchar
ntext	ntext	text
nvarchar	nvarchar	varchar
sql_variant	varchar	varchar

uniqueidentifier uniqueidentifier varchar

Note You will be required to change the default data type if you set the default data type to a supported data type while connected to a SQL Server 2000 database (bigint for example), then change your Access project connection to an earlier version of SQL Server.

- You tried to change a named default to a default constraint. Changing from a named default to a default constraint would permanently unbind the named default and is not allowed on Microsoft SQL Server.
- You tried to set a default value on a computed column. Computed columns cannot have default values.

[I'm not seeing all the records I expect.](#)

You may have set the [maximum record limit](#) too low for the Access project. This limit is designed to help improve performance when a user is viewing or browsing data in a datasheet or form and the query returns many records. If you are retrieving data from a select query, importing data from another Access project, or running an append or make-table query to copy data, you may want to set the maximum record limit to 0, which means no limit, to retrieve all records.

[The refresh interval does not appear to work in an Access project.](#)

The refresh interval is not intended for use with data from an OLE DB data source. In an Access database, the refresh interval (Point to the **Tools** menu, click the **Options** command, click the **Advanced Tab**, and select the **Refresh interval (sec)** option) is used to specify how often (in seconds) data is automatically updated in a multiuser environment. For example, if you type 120, the data is updated every 2 minutes. The underlying technology for this setting is Dynamic Data Exchange (DDE). However, Access projects use OLE DB to connect to a Microsoft SQL Server database. To refresh data manually in an Access project, use the **Refresh** command on the **Records** menu, or requery the records to retrieve up-to-date data.

[The data in my columns is suddenly truncated.](#)

You may have modified the column's data type or length and made it too small for the current data. For example, you changed the data type from nchar(100) to

nchar(80).

Change the data type or length back to the former value and restore the data from your backup file.

[Show All](#)

About updating data in an Access project

[General guidelines for updating data](#)

The following general guidelines apply when you are updating data in a [Microsoft Access project](#):

- Base tables in a query must have a primary key defined. When you first create a table in Database Designer or Table Designer, Access prompts you to create a primary key.
- To update any field (including the primary key field) in a single [table](#) or base table of a [join](#), you must have a unique field or combination of fields defined as a primary key constraint, a unique constraint, or any unique index on that table.
- To update data in a query, you must have UPDATE, INSERT and DELETE permission on the base table(s) referenced in the query, as well as SELECT permission on a view or EXECUTE permission on a [stored procedure](#) or user-defined function.

[When you can update data](#)

In general, you can update data under the following circumstances:

- In general, views can be updated subject to the query restrictions described in the next section below. However, in SQL Server 2000 databases, you can add the WITH VIEW_METADATA clause to the CREATE VIEW SQL statement (Select the **Update using view rules** check box in the **View** properties page). This clause allows updates to partitioned views, with certain restrictions, and the use of INSTEAD OF triggers to make the view updateable. For more information on updateable views, see the CREATE VIEW TRANSACT-SQL statement in the SQL Server documentation.
- A query based on a one-to-one relationship.

- A query based on a one-to-many relationship. However, there are important differences depending on the version of SQL Server.

SQL Server 2000 databases You can update any table in the join because Access automatically detects the unique table in the relationship (The "many" side of a join).

SQL Server 7.0 and 6.5 databases You can update data based on a [one-to-many](#) join in a form and page, but only for fields from the unique table. When you open a query that contains a join in Datasheet view, it is read-only, by default. However, you can update data from a form or if you set the form's **UniqueTable** property (see discussion below).

- A query based on a [self-join](#).
- A linked table based on a SQL Server Linked Server if the OLE DB data source allows.
- A UNION partitioned view (SQL Server 2000 only).

[When you can't update data](#)

In general, you cannot update data under the following circumstances:

- A query based on a [many-to-many](#) join.
- A query based on data marked as read-only in the database.
- A query that violates constraints on the base tables.
- A query that includes the DISTINCT keyword to exclude duplicate rows.
- A query based on a subquery that contains totals or aggregate functions.
- A form, datasheet, or page where the underlying SQL Statement of the [record source](#) contains a GROUP BY or COMPUTE clause.
- A linked table based on Transact SQL functions, because they create an ad hoc read-only connection to the external data source.
- In addition, you might not be able to update specific columns in the query results. The following list summarizes specific types of columns that you cannot update in the result set:

- Columns based on expressions (such as price * quantity AS extended_price).
- Rows or columns deleted by another user.
- Rows or columns locked by another user (locked rows can usually be updated as soon as they are unlocked).
- Timestamp or [image](#) columns.

[How Access displays updated and default column values in a datasheet or form](#)

[Display of updated column values](#)

When you are updating fields in a form, datasheet, or page, Microsoft Access can automatically insert a value for you, such as a default value, a calculated column, a timestamp value, an identity/AutoNumber field, or an action performed by a trigger. However, unlike a [Microsoft Access database](#), which in most cases updates the field value as soon as you leave the field, in an Access project the display of the value or effect of the operation may not occur until after you commit the record depending on the version of the database you are using and several property settings. Access may update or perform the operation at the server first and then refresh the current display on the client.

After you update a record, such as by calculating columns, changing the value of a joined field, or updating a foreign key field, Access will fix up the corresponding join tables. In a SQL Server 2000 database, Access normally resynchronizes the data on the form or datasheet to reflect the changed values. However, because this row fix-up requires additional round trips to the server, you may want to disable this setting with the **PerformResync** property. In SQL Server version 7.0 and 6.5 databases, this row fix up must be set by the form designer using the **ResyncCommand** property (see discussion below).

[Sorted and filtered results](#)

For SQL Server 2000 databases, when you update a sorted and filtered query, you don't lose the sort and filter.

For SQL Server 7.0 and 6.5 databases, Access removes filters and sorts and it appears as if the record disappears from its current position and moves to a new location.

[Display of default values](#)

Access 2000 databases Access displays default values when the form or datasheet displays new records by default. However, on a form, you can set the **Fetch Defaults** property to **No** to disable this for performance reasons.

Access 7.0 and 6.5 databases Access does not display default values when the form or datasheet displays new records. However, any defaults you have on controls will be displayed.

[Constraint validation and validation text](#)

Access does not inform a user about column constraint violations or display the Validation Text property value until the user attempts to save the record.

[Working with updatable snapshots in SQL Server 7.0 or 6.5 databases](#)

[Overview](#)

Access supports only inserting, deleting, and updating data in a row of a unique table (the "many" side of a join) in an [updatable snapshot](#) that has been created by a one-to-many join operation in a view, row-returning stored procedure, or SQL SELECT statement. Access doesn't allow you to update fields on the "one" side of the join.

A Form or page that is based on a one-to-many join must define the **UniqueTable** property in order for the [recordset](#) to be updatable. The unique table (also called the "most-many" table) is the "many" side of a one-to-many relationship. It is called a unique table because one record from the unique table corresponds to, at most, one record in the view, row-returning stored procedure, or SQL statement. For example, in a view that joins the Customers and Orders tables, Orders is the unique table because one record from the Orders table corresponds to, at most, one unique record in the view. By contrast, one record in the Customers table can correspond to multiple records in the view.

Additionally, for a [record source](#) that is a stored procedure or SQL statement containing unnamed parameters, if you want to display the current values of a record after it is updated, you must also set the **ResyncCommand** property. Note that if you don't set the **ResyncCommand** property in these cases, the only consequence is that you won't see current values in a record after an update or insert. The update or insert will still happen correctly. For other types of record sources, Access displays current values even if the **ResyncCommand** property is not set.

In an updatable snapshot, you see only current values in an edited or newly inserted record after it has been successfully saved (see discussion above). For example, the Orders table contains an identity column as its primary key. When you insert data into this table, the new OrderID value appears after you save the record or move to another record (which implicitly saves the record). Similarly, in a form based on a view joining Customers and Orders, if you update the OrderID field to a different CustomerID, the Address field will not show the address of the new customer until after the record has been saved.

Finally, all key fields or the UniqueTable must be in the SQL SELECT statement select list (although not necessarily visible). You must also select all NOT NULL columns that do not have server defaults defined and supply values for these columns in the recordset. This does not apply to fields which have the **Identity** property set.

[Recommended ways to update related data](#)

The following are recommended ways to update related data:

- AutoForm is a handy way to create a form that allows updating of a view or stored procedure based on a join. Use AutoForm to quickly create the form, then switch to design view to set the **UniqueTable** property. You can also change the **DefaultView** property if you prefer a Datasheet view to a single Form view.
- Displaying several "filtered" forms allows you to base one form on a main table and have buttons on this form that pop up other modal forms that are filtered from the data in the main form. For example, the main form could be based on "Customers" and you would have a "View Orders For Customer" button that would open a second form based on the Orders table that filters by the CustomerID field on the main form.

- To update all data in a one-to-many relationship, you can create a form/subform combination. For example, instead of displaying data on a single form that joins orders and customers (in this case, the customer fields would be read-only), you can create a form for customers and a subform for orders. Because a form/subform combination uses two different recordsets, you can update all fields. Additionally, you can display this form/subform combination as a datasheet with a subdatasheet.
- In a page, you can simulate a form/subform combination by linking two pages using a hyperlink field, and then use Microsoft FrontPage to create two frames, one for each page.

[Show All](#)

About securing an Access project (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Microsoft Jet [User-Level security](#) can't be used with a [Microsoft Access project \(.adp\)](#) file or .ade file. Users should help protect their copy of the Access project file using their computer's file system and periodically make a backup copy if they add their own forms or reports. If their copy of the Access project file is accidentally or unintentionally altered, users can recopy the file from their backup copy or its central location.

Securing an Access project

Strategies to help protect your project file and its objects are briefly described below and roughly arranged by levels of security.

[Show or hide objects in the Database window](#)

This security method is the least secure because it is relatively simple to show any hidden objects.

Note that because tables, views, database diagrams, and stored procedures reside in the SQL Server database, you cannot hide these objects in the database window.

[Using startup options](#)

You use the startup options to specify or determine settings such as which form opens automatically when your Access project file opens and your Access project application's title and icon. In a new Access project file, the startup properties do not exist until a user makes a change to the default settings in the **Startup** dialog box.

[Securing data access pages](#)

To help protect a data access page and the data it accesses, do the following:

- Help protect the Access project file containing the data access page's link, and the data access page's corresponding HTML file by using the file system security of the computer where you have stored these files.
- Help protect the database connected to the data access page by helping protect the database against unauthorized users, controlling the level of access to the data once a user logs in, and not saving the database password with the data access page.
- Help prevent unauthorized access from a malicious script by using Internet Explorer security options and using three-tier data access.

[Securing Visual Basic for Applications \(VBA\) code](#)

You can help protect Visual Basic for Applications (VBA) code in modules and modules behind forms and reports with a password which you enter once per session. The password helps prevent unauthorized users from editing, cutting, pasting, copying, exporting, and deleting VBA code.

To help protect your code, you can remove editable VBA code from your Access project file and help prevent modifications to the code behind your forms and reports by saving it as an ADE file.

Securing a Microsoft SQL Server database

[SQL Server database objects employing security technology](#)

You can help protect data and other database objects that are stored in an SQL Server database (tables, views, stored procedures, and database diagrams) by doing the following:

- Encrypt views, stored procedures and triggers. You can encrypt views to help prevent a user from changing the view. You can also encrypt stored procedures and triggers when you edit them by using Transact-SQL Statements (for example, `CREATE PROCEDURE...WITH ENCRYPTION`).

Important Once you encrypt a view, you can't change the design of that view. If you anticipate the need to revise an encrypted view, save the SQL statements in a text file in a secure location before you encrypt it. You can then delete the encrypted view, change the saved view, and re-encrypt it.

- Implement full security features using the SQL Server Enterprise Manager. See the SQL Server documentation.

[Help control logging on to SQL Server](#)

From a Microsoft Access project, you can administer basic security tasks by using the **Database Security** command to add, edit, or delete server logins, database users, and database roles. You must have SQL Server installed on the same computer as the Access project for these commands to work.

SQL Server provides two security methods to help validate logins to the SQL Server database:

- SQL Server security, which uses SQL Server user names and passwords.
- Windows integrated security, which uses Windows account user names. In this case, Windows authenticates a user when they log on, and SQL Server accepts Windows accounts without requiring a log on using other security measures.

[Show All](#)

About securing Visual Basic for Applications code in Access projects

[Helping protect VBA code](#)

In a [Microsoft Access project](#) (.adp), you can help protect all modules, and modules behind forms and reports, by using a password for [Visual Basic for Applications](#) (VBA) code, which you and other users must enter to view or edit VBA code in the [Visual Basic Editor](#). Once you set a password, you enter this password once per [session](#). The password is required not only for viewing and editing, but also for cutting, copying, pasting, exporting, and deleting any module. Note, however, that using these security measures to help protect your VBA code doesn't prevent you or other users from running existing VBA code.

[About .ade files](#)

If your Microsoft Access project contains Visual Basic for Applications (VBA) code, saving your Access project as an [.ade](#) file compiles all modules, removes all editable source code, and compacts the destination Access project file. Your VBA code will continue to run, but it cannot be viewed or edited.

Your Access project file will continue to function normally— you can still update data and run reports. Additionally, the size of your Access project will be reduced due to the removal of the code, so memory usage is optimized, which will improve performance.

Saving your Access project file as an .ade file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in [Design view](#).
- Adding, deleting, or changing references to [object libraries](#) or databases.
- Changing code— an .ade file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, data access pages, views, stored procedures, database diagrams, and macros can be imported from or exported to non-.ade Access projects.

[Considerations if you need to modify the design of forms, reports, or modules](#)

Be sure to save a copy of your original Access project file. If you need to modify the design of forms, reports, or modules in an Access project file that you have saved as an [.ade](#) file, you must modify the original Access project, and then resave it as an [.ade](#) file.

You won't be able to open, convert, or run code in an Access 2002 or later [.ade](#) file in future versions of Access. The only way to convert an Access 2002 or later [.ade](#) file to a future version will be to open the original Access project that the [.ade](#) file was created from, convert it to the later version of Access, and then save the converted Access project file as an [.ade](#) file.

[Considerations before saving your Access project file as an .ade file](#)

Some restrictions may prevent you from saving your Access project file as an [.ade](#) file:

- You must have password access to the VBA code if a password is defined.
- If your Access project references another Access project or [add-in](#), you must save all Access projects and add-ins in the chain of references as [.ade](#) or [.mde](#) files.

[About references and .ade files](#)

If you try to create an [.ade](#) file from an Access project file or an add-in that references another Access project or add-in, Access displays an error message and doesn't let you complete the operation. To save an Access project that references another Access project as an [.ade](#) file, you must save all projects in the chain of references as [.ade](#) or [.mde](#) files, starting from the first project referenced. After saving the first project as an [.ade](#) or [.mde](#) file, you must then update the reference in the next project to point to the new file before saving it as an [.ade](#) file, and so on.

For example, if Project1.adp references Project2.adp, which references Project3.mda, you would proceed as follows:

1. Save Project3.mda as Project3.mde.

2. Open Project 2.adp and change its reference to point to the new Project3.mde.
3. Save Project2.adp as Project2.ade.
4. Open Project1.adp and change its reference to point to the new Project2.ade.
5. Save Project1.adp as Project1.ade.

[About securing .ade files](#)

A Microsoft Access project can't use the same [user-level security](#) features that a [Microsoft Access database](#) (.mdb) can. The data and other objects stored on your [SQL Server](#) must employ the SQL Server's security features.

To help protect access to the VBA code in an Access project, you must password-protect its VBA project or remove your VBA source code by saving your Access project file as an .ade file. To help protect access to the design of forms and reports in an Access project file, you can set startup options or save your Access project file as an .ade file. The only way to help protect access to macros in an Access project file is by setting startup options.

[Show All](#)

Protect a Microsoft Access project (ADP)

Unlike a [Microsoft Access database](#), you can't protect forms, reports, or macros in an [Microsoft Access project](#) (.adp) by using [user-level security](#). To protect form and report objects, you can hide the objects in the Database window or set startup options. To protect access to the design of forms and reports in an Access project file, you can either set startup options or save your Access project file as an [.ade](#) file. To protect access to macros in an Access project file, use startup options. You can protect a data access page by using file or folder security for your operating system. You can also protect your [Visual Basic for Applications \(VBA\)](#) code by converting your Access project file to an .ade file or by setting a password. And finally, if your Access project is connected to a Microsoft SQL Server 6.5 (or later) database with SQL Server security implemented, then you can change the logon password from within Access.

[Show or hide database objects in the Database window](#)

[Specify whether a database object is hidden or visible](#)

1. Under **Objects** in the [Database window](#), click the type of database object whose properties you want to change.
2. Click **Properties**  on the toolbar.
3. Select or clear the **Hidden** check box.

Note In a [Microsoft Access project](#), you cannot change the properties of a table, query, or [database diagram](#) because these objects reside in the Microsoft SQL Server database. However, you can change the properties of a form, report, macro, or module, because these objects reside in the Access project itself, not in the Microsoft SQL Server database. You can also change the properties of a [data access page](#).

[Show or hide objects defined as hidden by default](#)

If you need to work with objects that are defined as hidden without changing

their status as hidden, you can display them in the [Database window](#).

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **Hidden objects** check box. Microsoft Access displays hidden objects with dimmed icons to distinguish them from objects that aren't defined as hidden.

[Show or hide system objects by default](#)

Microsoft Access automatically creates [system objects](#) when you create a new database.

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **System objects** check box.

[Use startup options](#)

You can use startup options to specify, for example, what form to display, whether toolbars can be customized, and whether [shortcut menus](#) are available in your [Microsoft Access file](#). You can also use a special [macro](#) named AutoExec to carry out an [action](#) or series of actions when your database first opens. When you open a database, Microsoft Access looks for a macro with this name and, if it finds one, runs it automatically.

1. On the **Tools** menu, click **Startup**.
2. Select the options or enter the settings you want to use.

[Create a macro that runs when an Access file first opens](#)

1. Create a [macro](#) containing the actions you want to run when you open the database.
2. Save the macro with the name AutoExec.

The next time you open the database, Microsoft Access runs this macro automatically.

Note If you don't want to run the AutoExec macro when you open the database,

hold down the SHIFT key when the database opens.

[Protect data access pages](#)

Data access pages are [Hypertext Markup Language \(HTML\)](#) files that reference the data in a database. The data access pages aren't actually stored in the [Microsoft Access file](#). For this reason, Access has no control over the security of the data access page files. To protect a data access page that is stored on a local or network file system, assuming you have the appropriate [permissions](#), you can use the following procedure.

Note To protect data access page files that are stored on an [HTTP](#) server, you must use the security features available on the server itself.

1. Open **Windows Explorer**  or **My Computer**.
2. Locate the folder where the data access page's HTML file is located. By default, the file will be located in the same directory as your Access database.
3. Right-click the data access page (.htm) file or the folder containing the file, click **Properties** on the shortcut menu, and then select the **Read-only** check box.

[Protect Visual Basic for Applications code](#)

You can protect the [Visual Basic for Applications \(VBA\)](#) code in your [Microsoft Access project \(.adp\)](#) by using a password or by saving a copy of your file as an [.ade](#) file. Saving your .adp file as an .ade file provides greater security than password protecting your code. However, there are some limitations in using an .ade file.

[Protect Visual Basic for Applications code with a password](#)

To prevent someone from viewing or making changes to your [Microsoft Visual Basic for Applications \(VBA\)](#) code, you can protect the code by requiring a password.

1. Open the [Microsoft Access project \(.adp\)](#) file or [Microsoft Access database \(.mdb\)](#) file that has the VBA code you want to help protect.
2. In the Database window, point to **Macro** on the **Tools** menu, and then click

Visual Basic Editor.



You can also press ALT+F11.

3. In the Microsoft Visual Basic Editor, click **<Access database or Access project name> Project Properties** on the **Tools** menu.
4. On the **Protection** tab, select the **Lock project for viewing** check box. If you set a password but don't select **Lock project for viewing**, the code can be viewed and edited by anyone, but security measures still help project the **Project Properties** dialog box.
5. In the **Password** box, type your password.



Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

6. In the **Confirm password** box, confirm your password by typing the password again, and then click **OK**.

The password is now set. The next time you or any other user opens the database, a dialog box will be displayed that requests a password.

Important If you forget your password, it can't be recovered, and you won't be able to view or edit the VBA code.

[Make an ADE file](#)

Cautions

Be sure to save a copy of your original Microsoft Access project (.adp) file.

- You can't modify the design of forms, reports, or modules in an Access project file saved as an .ade file. If you need to change the design of these objects, you must do so in the original Access project file and then resave the Access project file as an .ade file.
 - You can't create an .ade file in Access 2002 or later from an Access project file created in an earlier version. You can only save an .ade file from an Access project file from the same version of Access.
 - You also won't be able to convert an Access project file saved as an .ade file in future versions of Microsoft Access. You will be able to run the .ade file in a later version of Access.
1. Close the Access project file. If you are working in a [multiuser](#) environment, make sure that all other users have closed the Access project file.
 2. On the **Tools** menu, click **Database Utilities**, and then click **Make ADE File**.
 3. In the **Database To Save As ADE** dialog box, specify the Access project file you want to save as an .ade file, and then click **Make ADE**.
 4. In the **Save ADE As** dialog box, specify a name, drive, and folder for the Access project.

[Change the password of your Access project](#)

If your [Microsoft Access project \(.adp\)](#) is connected to a Microsoft SQL Server 6.5 (or later) database with SQL Server security implemented, then you can change the logon password from within Access.

1. Open an Access project file or switch to the [Database window](#) for the open project.
2. On the **Tools** menu, point to **Security**, and then click **Set Login Password**.
3. In the **Old Password** box, type the current password.
4. In the **New Password** box, type your password.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

5. In the **Verify** box, confirm your password by typing the password again, and then click **OK**.

The password is now changed and Access resets the .adp connection with the new password information. The next time you or any other user opens the project, a dialog box will be displayed that requests the new password.

Important If you lose or forget your password, it can't be recovered, and you won't be able to open your project.

[Show All](#)

Selecting the type of security validation used when logging on to SQL Server (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Note You must have SQL Server installed on the same computer as the Access project for these commands to work.

SQL Server provides two security methods for validating logins to the SQL Server database:

- SQL Server security, which uses SQL Server user names and passwords.
- Windows integrated security, which uses Windows account user names. In this case, Windows [authenticates](#) a user when they log on, and SQL Server accepts Windows authentication.

To select the type of security validation used, do one of the following:

[Use SQL Server security](#)

1. Select the **Use a specific username and password** checkbox on the **Connection** tab of the **Data Link Properties** dialog box.
2. Do one of the following:
 - Clear the **Allow saving of password** check box to require users to log on to the SQL Server database each time they open the Access project.
 - Select the **Allow saving of password** check box if you want to create an application that doesn't require a user to repeatedly log in to the SQL Server database.

Security Avoid saving logon information when connecting to data sources.

This information may be stored as plain text, and a malicious user could access the information to compromise the security of the data source.

[Use Windows authentication to connect an SQL Server database](#)

Select the **Use Windows NT Integrated Security** checkbox on the **Connection** tab of the **Data Link Properties** dialog box.

You don't need to provide a password because SQL Server uses your network login identity.

About enabling the System Administrator (SA) user name in an Access project

When the Microsoft SQL Server 2000 Desktop Engine (formerly called the Microsoft Database Engine or MSDE) is installed on Microsoft Windows computers, it is installed with Windows Authentication implemented (this feature is also known as integrated security). When a user connects to the SQL Server database through a Microsoft Access project, the connection is enabled through a Windows user account. SQL Server verifies that the account name and password were validated when the user logged on to the system and grants access to the database, without requiring a separate logon name or password.

With the default installation of the Microsoft SQL Server 2000 Desktop Engine with integrated security, the user must be a member of the administrator group for the machine on which the Access project resides. This makes it difficult to allow other users to gain access to the Access project.

There is currently no simple way for the system administrator to create new logon accounts to the locally installed SQL Server database except by using SQL Tools or Transact-SQL (TSQL) commands. See the Microsoft Developer Network Web site for more information on TSQL.

You can change the security mode for accessing the SQL Server database on a computer running Windows 2000 or later. By setting properties from the **View** menu, you can enable mixed mode security. Mixed mode security allows access to the database from either a Windows account or through a SQL Server account by using the default system administrator (SA) account with no password.

Using the default SA account, users can access the SQL Server database through the Access project without any additional security requirements. The administrator of the SQL Server can also add additional security by changing the default SA account password.

In order for the **Enable system administrator (SA) user name** check box to be

enabled, the following must be true.

- The Access project must be connected to a SQL Server 7.0 (or later) database or a Microsoft SQL Server 2000 Desktop Engine.
- The SQL Server database must be running on a Windows Server platform.
- The SQL Server must be running on the same computer as the Access project.
- The current Windows user must be an Administrator for the computer.
- The current SQL Server user must be a member of the System Administrator role.

Security Avoid using the built-in system administrator (**sa**) logon account. Instead, make system administrators members of the sysadmin fixed server role, and have them use their own accounts to log on. Use **sa** only when there is no other way to log on. To prevent unauthorized access through the **sa** logon account, you should assign that account a strong, unique password.

After changing the security mode, you can change the SA password by using the **Set Login Password** command (on the **Tools** menu, point to **Security**).

[Show All](#)

Enable or disable the System Administration (SA) user name in an Access project (ADP)

Security Avoid using the built-in system administrator (**sa**) logon account. Instead, make system administrators members of the sysadmin fixed server role, and have them use their own accounts to log on. Use **sa** only when there is no other way to log on. To prevent unauthorized access through the **sa** logon account, you should assign that account a strong, unique password.

[Enable the SA user name](#)

If your Microsoft Access project is connected to a locally installed SQL Server 7.0 (or later) database or a Microsoft SQL Server 2000 Desktop Engine (formerly called MSDE) with Microsoft Windows Authentication implemented, you can enable the system administrator (SA) user name with a blank password so that other users can access the application. Enabling the SA user name will also allow the System Administrator to change the SA password for additional security.

Note In order for the **Enable system administrator (SA) user name** check box to be enabled, the following must be true:

- The Access project must be connected to a SQL Server 7.0 (or later) database or a Microsoft SQL Server 2000 Desktop Engine (formerly called MSDE).
- The SQL Server database must be running on a Windows Server platform.
- The SQL Server must be running on the same computer as the Access project.
- The current Windows user must be an Administrator for the computer.
- The current SQL Server user must be a member of the System

Administrator role.

1. Open an Access project file or switch to the [Database window](#) for the open project.
2. On the **View** menu, click **Server Properties**.
3. Select the **Enable system administrator (SA) user name** check box.
4. Click **OK** in the message box.

When this option is selected, the following actions take place.

- The Access project is disconnected from the SQL Server database. This closes all open designers (such as the View Designer).
- The login mode is changed to mixed mode security (both SQL Server security and Windows Authentication are implemented).
- Access stops and starts the database service.
- The connection to the Access project is changed to use the SA user name with no password.
- The Access project is then reconnected to the database.

Notes

- It is strongly recommended that the SA password be changed by using the **Set Login Password** command (on the **Tools** menu, point to **Security**).
- If a connection to the Access project fails, a standard SQL Server login dialog box is displayed so that the user name or password can be corrected.

[Disable the SA user name](#)

1. Open a Microsoft Access project file or switch to the [Database window](#) for the open project.
2. On the **View** menu, click **Server Properties**.
3. Clear the **Enable system administrator (SA) user name** check box.

When this option is cleared, the following actions take place.

- The Access project is disconnected from the SQL Server database which closes all open designers (such as the View Designer). See the Microsoft Developer Network Web site for more information on designers.
- The login mode is changed to Microsoft Windows Authentication only.

- Access stops and starts the database service.
- The connection to the Access project is changed to use Trusted Security. See the Microsoft Developer Network Web site for more information on Trusted Security.
- The Access project is then reconnected to the database.

Change an encrypted view in an Access project (ADP)

You can encrypt views to help prevent a user from changing the view. However, once you encrypt a view, you can't change its design. If you anticipate the need to revise an encrypted view, save the SQL in a text file in a controlled location before you encrypt it.

To change an encrypted view

1. Delete the old encrypted view.
2. Create a new view with the same name.
3. Cut and paste the SQL statement from the saved text file into the SQL pane of the new view.
4. Revise the new view as needed.
5. Save the revised SQL statement in a new text file in a controlled location.
6. Encrypt the new view. Additional information on encrypting views can be found in Microsoft SQL Server Help.

[Show All](#)

About ADE files (ADP)

If your [Microsoft Access project \(.adp\)](#) file contains [Visual Basic for Applications \(VBA\)](#) code, saving your Access project as an [.ade](#) file compiles all modules, removes all editable source code, and compacts the destination Access project file. Your VBA code will continue to run, but it cannot be viewed or edited.

Your Access project file will continue to function normally— you can still update data and run reports. Additionally, the size of your Access project will be reduced due to the removal of the code, so memory usage is optimized, which will improve performance.

Saving your Access project file as an [.ade](#) file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in [Design view](#).
- Adding, deleting, or changing references to [object libraries](#) or databases.
- Changing code— an [.ade](#) file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, data access pages, views, stored procedures, database diagrams, and macros can be imported from or exported to non-[.ade](#) Access projects.

[Considerations if you need to modify the design of forms, reports, or modules](#)

Be sure to save a copy of your original Access project file. If you need to modify the design of forms, reports, or modules in an Access project file that you have saved as an [.ade](#) file, you must modify the original Access project, and then resave it as an [.ade](#) file.

You won't be able to open, convert, or run code in an Access 2002 or later [.ade](#) file in future versions of Access. The only way to convert an Access 2002 or later [.ade](#) file to a future version will be to open the original Access project that the [.ade](#) file was created from, convert it to the later version of Access, and then save the converted Access project file as an [.ade](#) file.

[Considerations before saving your Access project file as an .ade file](#)

Some restrictions may prevent you from saving your Access project file as an .ade file:

- You must have password access to the VBA code if a password is defined.
- If your Access project references another Access project or add-in, you must save all Access projects and add-ins in the chain of references as .ade or .mde files.

[About references and .ade files](#)

If you try to create an .ade file from an Access project file or an add-in that references another Access project or add-in, Access displays an error message and doesn't let you complete the operation. To save an Access project that references another Access project as an .ade file, you must save all projects in the chain of references as .ade or .mde files, starting from the first project referenced. After saving the first project as an .ade or .mde file, you must then update the reference in the next project to point to the new file before saving it as an .ade file, and so on.

For example, if Project1.adp references Project2.adp, which references Project3.mda, you would proceed as follows:

1. Save Project3.mda as Project3.mde.
2. Open Project 2.adp and change its reference to point to the new Project3.mde.
3. Save Project2.adp as Project2.ade.
4. Open Project1.adp and change its reference to point to the new Project2.ade.
5. Save Project1.adp as Project1.ade.

Summary of dialog box Help in Microsoft Access Project Designers

[Add Table Dialog Box \(Database Designer\)](#)

[Add Table Dialog Box \(Query Designer\)](#)

[Choose Name Dialog Box](#)

[Choose Target Table for Insert Results Dialog Box](#)

[Choose Target Table for Insert Values Dialog Box](#)

[Column Selection Dialog Box](#)

[Create Relationship Dialog Box](#)

[Database Changes Detected Dialog Box \(Database Designer\)](#)

[Datatype Change Required Dialog Box](#)

[Define Column Collation Dialog Box](#)

[Delete Table Dialog Box](#)

[Indexes Dialog Box](#)

[Make-Table Dialog Box](#)

[Page Setup Dialog Box](#)

[Post-Save Notifications Dialog Box](#)

[Query Definitions Differ Dialog Box](#)

[Save Dialog Box](#)

[SQL Syntax Errors Encountered Dialog Box](#)

[Update Table Dialog Box](#)

[Validation Warnings Dialog Box](#)

Summary of extended properties in a Microsoft Access Project

[Check Constraints Property Page](#)

[Columns Property Page](#)

[Data Property Page](#)

[Function Parameters Property Page](#)

[General Property Page](#)

[Indexes/Keys Property Page](#)

[In-Line Function Property Page](#)

[Join Line Property Page](#)

[Lookup Property Page](#)

[Query Property Page](#)

[Relationships Property Page](#)

[Stored Procedure Parameters Property Page](#)

[Stored Procedure Property Page](#)

[Tables Property Page](#)

[View Property Page](#)

[Show All](#)

Summary of messages in Microsoft Access Project Designers

All messages for the user interface components in a [Microsoft Access project](#), including Database Designer, Table Designer, Query Designer, Query Builder, and SQL Text Editor are listed below in alphabetical order.

Note Additional messages are documented in the Microsoft SQL Server Books Online help information.

[<0s> cannot be used in this query type.](#)

[<0s> in expression is not part of the query.](#)

[<0s> support not available in this server version.](#)

[A blank constraint expression is not allowed.](#)

[A database diagram with that name already exists in the database.](#)

[A number of rows were affected.](#)

[A pair of matching columns is required to create a relationship.](#)

[A primary key can't be created on column '<0s>' because it allows null values.](#)

[A primary key or index cannot be created on columns with a datatype of <0s>.](#)

[A primary key or index cannot have more than <0d> columns.](#)

[A primary key or unique constraint must be defined for table '<0s>' before it can participate in a relationship.](#)

[A relationship cannot contain more than '<0d>' columns.](#)

[Adding a formula to the selected column will delete all indexes and relationships it participates in.](#)

[Adding a formula to the selected column will delete all relationships it participates in as a foreign key.](#)

[ALIAS name is already being used.](#)

[ALL cannot be used with CUBE or ROLLUP.](#)

[Ambiguous outer join \(+\) operator.](#)

[An expression cannot be used as a parameter value.](#)

[An index can't exist on a blank column.](#)

[Another user modified this database diagram while you were working on it.](#)

Appropriate SQL cannot be generated.

Are you sure you want to delete the selected relationship from your database?

Are you sure you want to permanently delete table '<0s>' from your database?

Are you sure you want to permanently delete the selected tables from your database?

Asterisk (*) may not be used in an indexed view.

Bad top value.

Both sides of a relationship must have the same number of columns.

Cannot assign alias to this field.

Cannot convert entry to valid date/time.

Cannot convert to proper type.

Cannot delete rows with unknown keys.

Cannot edit rows with unknown keys.

Cannot edit this cell.

Cannot filter this expression.

Cannot insert into this expression.

Cannot open encrypted <0s> <1s>.

Cannot put expression on select list.

Cannot update this expression.

Cannot use column whose data type is image or text in this context.

Cannot use long data type in this context.

Changing a column data type results in an index that is too large.

Changing the column to the selected data type will delete its indexes and all relationships it participates in.

Changing the column to the selected data type will delete its indexes.

Changing the column to the selected data type will delete relationships.

Changing this alias will result in a duplicate output column name.

Collate clause may not be used in an indexed view.

Column '<0s>' is a computed column, and cannot contribute to foreign keys.

Column '<0s>' is a computed column, and cannot participate in indexes or relationships.

Column '<0s>' is used to enforce the full-text key on table <1s> and must be <2d> bytes or less.

Column '<0s>' participates in full-text indexing on table '<1s>'. After this change, the column will no longer participate in the table's full-text index.

Column '<1s>' in table '<0s>' participates in index '<2s>'. Columns with a

datatype of '<3s>' cannot participate in indexes or primary keys.

Column cannot be updated.

Column has no name and will be deleted.

Column is the full-text filter for columns that participate in full-text indexing on a table.

Columns in this expression have incompatible collations.

Comment exceeds n bytes.

Connection failure.

Data source alias is read only.

Data source base name is read only.

Data type error in expression.

Data type mismatch - no conversion possible.

Data type mismatch - use the Transact SQL CONVERT function.

Database Designer Error

Database Server Version Runtime Error.

Default Value property cannot be set on column '<0s>' because it is a computed column.

DELETE statement conflicted with COLUMN REFERENCE constraint.

Deleting the selected columns will also delete relationships.

Do you want to save changes in this database diagram?

Do you want to save changes in this table?

Do you want to suppress further error messages telling you why records can't be pasted?

Editing this default will permanently unbind the named default.

Edits not allowed. HAVING clause not allowed in an indexed view.

Enter an expression in the Column cell first.

Error after function '<0s>'

Error after predicate near '<0s>'

Error before EXISTS clause.

Error before EXISTS clause: '<0s>' not recognized.

Error before FROM clause.

Error following UNION operator.

Error in column list.

Error in column list: '<0s>' not recognized.

Error in destination table specification.

Error in FROM clause near '<0s>'

Error in GROUP BY clause near '<0s>'

Error in GROUP BY clause.

Error in HAVING clause near '<0s>'.

[Error in HAVING clause.](#)
[Error in INSERT statement.](#)
[Error in INSERT statement: '<0s>' not recognized.](#)
[Error in join expression.](#)
[Error in join expression: '<0s>' not recognized.](#)
[Error in list of function arguments: '<0s>' not recognized.](#)
[Error in list of values in IN clause.](#)
[Error in list of values in IN clause: '<0s>' not recognized.](#)
[Error in list of values.](#)
[Error in list of values: '<0s>' not recognized.](#)
[Error in ON clause near '<0s>'.](#)
[Error in optional FROM clause.](#)
[Error in ORDER BY clause near '<0s>'.](#)
[Error in ORDER BY clause.](#)
[Error in SELECT clause: alias '<0s>' not recognized.](#)
[Error in SELECT clause: alias not recognized.](#)
[Error in SELECT clause: expression near '<0s>'.](#)
[Error in set list in UPDATE clause.](#)
[Error in table name or view name in DELETE clause.](#)
[Error in table name or view name in DELETE clause: '<0s>' not recognized.](#)
[Error in table name or view name in INSERT clause.](#)
[Error in table name or view name in INSERT clause: '<0s>' not recognized.](#)
[Error in table name or view name in UPDATE clause.](#)
[Error in text following query statement: '<0s>'.](#)
[Error in values list in INSERT INTO clause.](#)
[Error in values list in INSERT INTO clause: '<0s>' not recognized.](#)
[Error in WHERE clause near '<0s>'.](#)
[Error loading table.](#)
[Error validating constraint.](#)
[Error validating the default for a column.](#)
[Error validating the formula for column.](#)
[Function argument count error.](#)
[Identity column '<0s>' in table '<1s>' must have a datatype of int, smallint, tinyint, decimal or numeric with scale of 0.](#)
[Identity seed must be a number containing <0d> digits or less.](#)
[If you save the view encrypted, you will no longer be able to alter the view definition.](#)
[Ignoring illegal use of ALL.](#)

[Ignoring ODBC syntax.](#)

[Illegal sequence use.](#)

[Illegal use of collation clause.](#)

[Illegal use of expression list.](#)

[Illegal use of outer join operator.](#)

[Incomplete column list.](#)

[Incomplete parameters list.](#)

[Incomplete parameters or column list.](#)

[Incomplete SET clause.](#)

[Incomplete VALUES list.](#)

[Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be <2d> bytes or less.](#)

[Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be single-column.](#)

[Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be unique.](#)

[Index '<0s>' is used to enforce the full-text key on table '<1s>' and must not be null.](#)

[Index already exists.](#)

[Index must have at least one column.](#)

[Indexed View must contain a clustered index.](#)

[Internal Error.](#)

[Invalid data type.](#)

[Invalid entries must be resolved before you can exit this pane.](#)

[Invalid escape character.](#)

[Invalid fill factor; enter an integer between 0 and 100.](#)

[Invalid identifier '<0s>'.](#)

[Invalid name. You must provide a name for this object.](#)

[Invalid or missing expression near '<0s>'.](#)

[Invalid or missing expression.](#)

[Invalid prefix or suffix characters.](#)

[Invalid row index: Goto failed.](#)

[Invalid text or symbol.](#)

[Invalid view name.](#)

[IS operator can only be used with NULL or NOT NULL.](#)

[Linked Server object cannot be used as a destination with this query type.](#)

[Missing escape character in LIKE predicate.](#)

[Missing FROM clause near '<0s>'.](#)

[Missing FROM clause.](#)

Missing FROM keyword near '<0s>'.

Missing FROM keyword.

Missing INTO keyword near '<0s>'.

Missing INTO keyword.

Missing or incomplete SELECT clause.

Missing pattern in LIKE predicate.

Missing SET keyword.

Missing subquery correlation name.

Missing subquery or the operator you entered requires parenthesis.

Missing subquery.

Must enter either TRUE or FALSE.

Need alias for this field.

NOT cannot be used in a column cell.

Object name cannot begin with #.

Object with name already exists in the database.

ODBC driver returned an invalid ODBC version and needs to be updated.

One or more selected tables are already on the database diagram.

Only one clustered index can be created on table '<0s>'.

Only one ROWGUID column is allowed per table.

OpenRowset cannot be used as a destination with this query type.

ORDER BY not supported for CREATE TABLE query.

Outer join column may not be used with an IN predicate or subquery.

Poorly formed comment.

Poorly formed Transact SQL CAST function.

Primary key or index cannot be created on a column with no name.

Primary key or index cannot be created on column '<0s>' because its data type is '<1s>'.

Property cannot be modified.

Query Designer Error.

Query Designer supports no more than 1 table for this type of query.

Query has executed successfully.

Relationship '<0s>' was modified or deleted since the database diagram was loaded.

Removing the formula from the selected column leaves it with data type that is disallowed in indexes and relationships.

Row limit value should be between <xxx> and <xxx>.

Rows pasted.

Select both primary key table and foreign key table before selecting any field for the relationship.

Setting for Length must be from <0d> to <1d>.

Setting for Precision must be from <0d> to <1d>.

Setting for Scale must be from <0d> to <1d>.

SQL statement could not be parsed.

SQL text cannot be represented in the grid pane and diagram pane.

SQL Verify failed.

Summary of messages in Access Project Designers

Syntax error in table reference.

Syntax error in table reference: '<0s>' not recognized.

Table '<0s>' already exists.

Table '<0s>' has no columns.

Table '<0s>' is marked for deletion and was not added to the diagram.

Table '<0s>' no longer exists in the database.

Table is not in the query definition.

Table name cannot include the double-quote character (").

The alias name is too long.

The Allow Nulls property can't be set on a column that is part of the primary key.

The Allow Nulls property can't be set on column '<0s>' because it is an identity column.

The collation properties of columns <0s> and <1s> do not match.

The columns in table '<0s>' do not match an existing enabled primary key or UNIQUE constraint.

The columns in table '<0s>' do not match an existing primary key or UNIQUE constraint.

The database row you are modifying no longer exists in the database.

The datatype of column '<1s>' in the '<0s>' table can't be changed because it participates in index '<2s>'.

The Default Value property can't be set on column '<0s>' because it is an identity column.

The existing relationship must have at least one pair of related columns.

The following datatype or size property of '<0s>.<1s>' doesn't match '<2s>.<3s>'.

The following error was encountered while changing the collation: <0s>.

The following schema-bound objects will be modified.

The identity increment must be a number containing <0d> digits or less.

The Identity property can't be set on column '<0s>' because it allows null values.

The Identity property can't be set on column '<0s>' because it has a default

value.

The index is used to enforce the full-text key for this table. Deleting this index will disable full-text indexing for the table. Do you want to proceed?

The name is too long.

The new relationship must have at least one pair of related columns.

The NOT keyword may not be used in a column cell.

The number of selected columns exceeds the number of columns in the target table.

The primary key or UNIQUE constraint cannot be changed until its existing relationships are deleted.

The primary key or Unique constraint cannot be changed while relationships to the existing primary key or Unique constraint are enforced.

The Query Designer does not support the <0s> SQL construct.

The Query Designer does not support the critical ODBC APIs.

The Query Designer does not support the current ODBC API.

The specified OR group number is already in use.

The SQL Server does not meet minimum requirements of version 6.5 service pack 3.

The SQL syntax has been verified against the data source.

The table being loaded into memory has a user-defined data type that is not recognized.

The table must have at least one column that is not computed.

The table-valued function used as target is not an in-line function.

The total size of an index or primary key cannot exceed 256 bytes.

The value for the extended property is too long.

The value you entered is not consistent with the data type or length of the column.

There are no columns selected. Please select one or more columns and re-run the query

There are not enough columns to match the value list.

There are not enough values in the subquery select list.

There are too many values from the sub-query select list.

There is no unique table in this query.

This backend version is not supported to design database diagrams or tables.

This cell contains the text string "<NULL>" which may not be processed correctly.

This operator cannot be used with columns with data type "uniqueidentifier".

This SQL statement type cannot be used in a view.

Too many characters for field width.

Too many or too few parameters specified.

Unable to execute create script because it does not begin with CREATE or ALTER.

Unable to execute create script because it does not begin with CREATE.

Unable to execute the create script; the table name is invalid or missing.

Unable to locate data source.

Unable to parse expression.

Unable to parse query text.

Unable to parse statement.

Unable to read object text; the text is encrypted.

Unable to save; the object name is invalid or missing.

Unexpected object type.

Unknown column.

Unknown conversion specification.

Unquoted alias contains white space.

Use of GROUP BY function in WHERE clause not allowed.

Use of Transact SQL CONVERT function might be unnecessary.

Values for '<0s>' must lie within the range <1d> to <2d>.

Values for property must lie within the range x to y.

View already exists in the database.

View definition includes no output columns or no items in the FROM clause.

View has indexes. If you remove schema binding, the indexes will be dropped.

View name cannot begin with #.

Warning: It is likely that the view definition will result in a view that cannot be indexed.

Warning: It is likely that your modification will result in a view that cannot be indexed.

WITH TIES clause requires an ORDER BY clause.

You are about to delete a row or rows.

You are about to paste n rows.

You are not logged in as the database owner or as a user that is a member of the db_owner role. You will not be able to save changes to tables that you do not own.

You cannot create a view which is self-referenced.

You entered <0d> characters for '<1s>'. The maximum number of

characters allowed is <2d>.

You entered x characters for a property; the maximum allowed is y.

You might not have permission to perform this operation, or the object <0s> might no longer exist in the database.

Your database diagram will be updated with the following changes to match the database before the following tables can be loaded.

Your entry cannot be converted to a valid date time value.

Your Logon does not have CREATE TABLE permission; you might not be able to make certain edits that require this permission.

Your table will be updated with the following changes to match the database.

About filters

[About filters \(MDB\)](#)

[About filters \(ADP\)](#)

About indexes

[About indexing fields and records in an Access database \(MDB\)](#)

[About indexes \(ADP\)](#)

About keys

[About primary keys \(MDB\)](#)

[About keys \(ADP\)](#)

About queries

[About types of queries \(MDB\)](#)

[About select and crosstab queries \(MDB\)](#)

[About action queries \(MDB\)](#)

[About creating queries \(ADP\)](#)

About sorting records

[About sorting records \(MDB\)](#)

[About sorting records \(ADP\)](#)

About table relationships

[About relationships in an Access database \(MDB\)](#)

[About table relationships \(ADP\)](#)

About tables

[About tables \(MDB\)](#)

[About tables \(ADP\)](#)

Add columns to a query

[Add or remove a field in the design grid \(MDB\)](#)

[Add columns to a query \(ADP\)](#)

Apply a filter

[Apply a filter \(MDB\)](#)

[Apply a filter \(ADP\)](#)

Backup and restore an Access file

[Back up and restore an Access database \(MDB\)](#)

[Backup and restore a database or Access project \(ADP\)](#)

Create a delete query

[Create a delete query \(MDB\)](#)

[Create a delete query \(ADP\)](#)

Create a filter

[Create a filter \(MDB\)](#)

[Create a filter \(ADP\)](#)

Create a join in a query

[Enable or disable automatic joins in queries \(MDB\)](#)

[Join tables and queries in a query \(MDB\)](#)

[Create a join between tables on multiple columns in a query \(ADP\)](#)

[Create a join between tables, views, or functions in a query manually \(ADP\)](#)

Create a relationship between tables

[Define relationships between tables \(MDB\)](#)

[Create a relationship between tables \(ADP\)](#)

Create an append query

[Create an append query \(MDB\)](#)

[Create an append query \(ADP\)](#)

Create an index

[Create an index to find and sort records faster \(MDB\)](#)

[Create an index \(ADP\)](#)

Create an input mask

[Create an input mask \(MDB\)](#)

[Create an input mask \(ADP\)](#)

Create an update query

[Create an update query \(MDB\)](#)

[Create an update query \(ADP\)](#)

Create and use a parameter query

[Create a parameter query \(MDB\)](#)

[Create and use a parameter query \(ADP\)](#)

Delete a join in a query

[Delete a join line in a query \(MDB\)](#)

[Remove a join in a query \(ADP\)](#)

Delete a relationship

[Delete a relationship \(MDB\)](#)

[Delete a relationship \(ADP\)](#)

Delete a sort

[Delete a sort \(MDB\)](#)

[Delete a local sort \(ADP\)](#)

Delete an index

[Delete an index \(MDB\)](#)

[Delete an index \(ADP\)](#)

Delete or remove a filter

[Delete or remove a filter \(MDB\)](#)

[Delete or remove a filter \(ADP\)](#)

Exclude duplicate rows in a query

[Prevent showing duplicate records in a query \(MDB\)](#)

[Exclude duplicate rows in a query \(ADP\)](#)

Modify a filter

[Modify a filter \(MDB\)](#)

[Modify a filter \(ADP\)](#)

Modify a join in a query

[Set or change the join type \(MDB\)](#)

[Change the type of join in a query \(MDB\)](#)

[Modify a join operator in a query \(ADP\)](#)

Run a query

[Run a query \(MDB\)](#)

[Execute a query in the Query Designer \(ADP\)](#)

Sort records

[Sort records \(MDB\)](#)

[Sort records \(ADP\)](#)

Total records in a query

[Total records in a query \(MDB\)](#)

[Count rows in a query \(ADP\)](#)

Troubleshoot filters

[Troubleshoot filters \(MDB\)](#)

[Troubleshoot filters \(ADP\)](#)

Troubleshoot queries

[Troubleshoot queries \(MDB\)](#)

[Troubleshoot queries \(ADP\)](#)

Troubleshoot sort

[Troubleshoot sort \(MDB\)](#)

[Troubleshoot sort \(ADP\)](#)

[Show All](#)

Changes made to your database when you use replication (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

When you replicate a Microsoft Access database, a number of changes are automatically made to your database.

[Fields that are added to your tables when you replicate a database](#)

When you replicate a database, Microsoft Access adds the following system fields to each replicated table in the replicated database:

Field	Description
s_GUID	A globally unique identifier for each record.
s_Lineage	A binary field that contains information about the history of changes to each record.
s_Generation	A field that stores information regarding groups of changes.

Notes

- The s_GUID, s_Lineage, and s_Generation system fields are system objects that may or may not be visible in your tables. To make these fields visible in replicated tables, click **Options** on the **Tools** menu, click the **View** tab, and then select the **System Objects** check box. To see the fields, open a replicated table.
- Additional fields (named Gen_*Fieldname*) are added for every Memo data type or OLE Object data type field in a table.
- You can't change the information stored in the system fields in a replicated table.

[Tables that are added to your database when you replicate it](#)

When a Microsoft Access database is replicated and certain conditions occur, new system tables may be added to the database. Of these, the following new

system tables may be of interest:

Table	Description
MSysSidetables	This table exists only if a conflict has occurred between the user's replica and another replica in the replica set. This table isn't replicated. It's supplied for information only, and its contents can be modified or deleted by custom conflict-resolution routines or by the user. All side tables are named <i>table_conflict</i> , where <i>table</i> is the original name of the table.
MSysSchemaProb	This table is present only when an error has occurred while updating the design of a replica. It provides additional details about the cause of the error. This is a local table and isn't replicated.
MSysReplicas	This table stores details, such as Replica ID and path, of all known replicas in the replica set. This table appears in all members of the replica set, but it is not replicated.
MSysTransAddress	This table stores addressing information for the Synchronizer and defines the set of synchronizers known to this replica set. This replicated table appears in all members of the replica set.
MSysTombstone	This table stores information about deleted records, and allows deletions to be dispersed to other replicas during the synchronization process. This table appears in all members of the replica set, but it is not a replicated table.
MSysRepInfo	This table stores information about the entire replica set, including the identity (GUID) of the Design Master. The table contains a single record. This replicated table appears in all members of the replica set.
MSysExchangeLog	This table stores information about replica synchronizations that have taken place. This is a local table and isn't replicated.

Notes

- These tables are visible when system objects are visible.
- You can't change the information stored in the MSysSchemaProb, MSysExchangeLog, MSysReplicas, MSysTransAddress, MSysTombstone, or MSysRepInfo system tables in a replicated database.

[Properties that are added to your database when you replicate it](#)

When you replicate a database or specify a database object as local, the following new properties may be added to the **Properties** collection of objects:

Property	Description
Replicable or ReplicableBool	A database or object property. When the property is set to T (or True for ReplicableBool), it indicates that the database or the table or query is now replicable. The Replicable and ReplicableBool properties may be used interchangeably.
KeepLocal	A property appended to a table or query. When the property is set to T , it indicates that the object should not be replicated when the database is replicated. An object that is already replicated can't have its KeepLocal property set to T .
ReplicaID	A property that provides each member of the replica set with unique identification. This property is read-only and is stored in the MSysReplicas system table.
DesignMasterID	The ReplicaID of the Design Master. It is stored in the MSysRepInfo system table as SchemaMaster.
ColumnLevelTracking	A database or table property. When the property is set to True (default), it indicates that conflicts are tracked at the column level of a table.
Replication ConflictFunction	The property is used to replace the Microsoft Access Conflict Viewer with a customized procedure that assists users in resolving synchronization conflicts.

Notes

- Microsoft Access objects (forms, reports, macros, and modules) are either all replicated or are all local. Therefore, the **Replicable** and **KeepLocal** properties apply only to tables and queries.
- To learn about the **Replicable**, **KeepLocal**, **ReplicaID**, and **DesignMasterID** properties, open the Visual Basic Editor and then open Visual Basic Editor Help.

[Changes to the behavior of AutoNumber fields when you replicate a database](#)

When you replicate a database, any incremental AutoNumber fields in your tables change to random numbering. All AutoNumber fields in existing records retain their values, but AutoNumber values for inserted records are random. In other words, the record numbers don't reflect the order in which the records were inserted and, consequently, the record inserted last doesn't necessarily have the highest value.

Before you replicate a database, determine whether any of your applications or users rely on an AutoNumber field's incremental numbering. If so, you can use an additional Date/Time field to provide sequential ordering information.

Note Replicas in an Access database that were created from a Microsoft SQL Server publication retain the AutoNumber functionality to support SQL Server's Named Range feature. You must manage these ranges in each replica.

[Additional size limitations on a replicated database](#)

The addition of three fields (s_GUID, s_Lineage, and s_Generation) imposes two limitations on your replicated tables:

- The Microsoft Jet database engine allows a maximum of 2,048 bytes (not counting Memo or OLE Object fields) in a record. Replication uses a minimum of 54 bytes to store unique identifiers, indexes, and information about changes to the record. If the record contains either Memo or OLE Object fields, replication uses an additional 4 bytes for each of those fields. The total number of bytes available in a record in a replicated table can be calculated as follows:

2,048 bytes

- 54 bytes for replication overhead

- (4 bytes * the number of Memo fields)

- (4 bytes * the number of OLE Object fields)

= the maximum number of bytes available

- The Microsoft Jet database engine allows a maximum of 255 fields in a table. Besides the three additional system fields, replication adds an extra

field for each Memo and OLE Object field. The total number of fields available in a replicated table can be calculated as follows:

255 fields

- 3 system fields

- the number of Memo and OLE Object fields

= the number of fields available

Few well-designed applications use all the available fields in a table or all the bytes in a record. However, if you have a large number of Memo or OLE Object fields in your table, you might run into constraints.

Note In addition to setting limits on the available number of characters and fields, the Microsoft Jet database engine also limits the number of nested transactions allowed. You can have a maximum of seven nested transactions in a nonreplicated database, but a replicated database can have a maximum of six nested transactions.

The addition of several new system tables also adds to the size of your database. Most of these new tables contain only a few records, but some of the new tables can grow significantly, depending upon the frequency of synchronization between replicas.

The size of your database file is significant for two reasons:

- The Microsoft Jet database engine supports files up to a maximum of 1 gigabyte in size, but many users have limited space available on their hard disks. A replicated database that is larger than the original database file might use up a significant portion of the available disk space.
- If you choose to make a backup copy of the nonreplicable form of your database, you must have sufficient space on your hard disk for both the replicated database and the backup copy.

[Show All](#)

About choosing a tool for replication (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

You can use one of the following methods for [replicating](#) and [synchronizing](#) databases:

[Briefcase replication](#)

Briefcase replication is perhaps the easiest way to implement replication and is a convenient tool for keeping a database file up-to-date. You can replicate the database on your laptop by using Briefcase replication, disconnect the laptop from the network, and then make changes to the data in the [replica](#). When you are finished working on the files on the laptop computer, you can then reconnect to the network, and automatically synchronize any changes between the database on your laptop and the [Design Master](#) on the network.

[Replication commands on the Microsoft Access **Tools** menu](#)

Microsoft Access provides replication menu commands that you can use while working in your databases. By using the **Replication** command on the **Tools** menu, you can:

- Create a [replica](#).
- Synchronize your replica with another member of the [replica set](#).
- Resolve synchronization conflicts by using the Conflict Resolver.
- Recover the replica set's [Design Master](#), if necessary.

[Replication in a Microsoft Access project](#)

A [Microsoft Access project](#) file contains several wizards to help you create a

[publication](#) and [subscription](#). Access projects support push, pull, and bi-directional synchronization.

In an Access project, you replicate the data, not the forms, reports, and other objects that are contained in the project. And an Access project cannot contain a [Microsoft Access database replica](#).

[Implementing replication programmatically](#)

Use [Jet and Replication Objects \(JRO\)](#) if you require programmatic control over the exchange of data and design information among members of the [replica set](#) in [Microsoft Access databases](#) (.mdb files only). For example, you can use JRO to write a procedure that automatically synchronizes a user's [replica](#) with the rest of the set when the user opens the database. To replicate a database programmatically, the database must be closed.

If your database was created with Microsoft Access 97 or earlier, you must use [Data Access Objects \(DAO\)](#) to programmatically replicate and synchronize it.

[Web server replication](#)

Web server replication is a tool you can use to work on files even when you are no longer connected to a Web server.

Microsoft Office 2003 users can continue to work on shared documents even when they log off the network. When the Web Publishing feature is installed, a replication mechanism automatically maintains a list of recently opened Web server documents and stores copies of them in a cache on the local hard disk. To add files to the cache, users select a file in Microsoft Windows Explorer, and then use menu options to make the files available offline.

A user can open and edit replicated files while offline. Then, the next time the user logs on to the network, all changes the user has made are automatically replicated to the server. Also any changes that have occurred on the server while the user has been offline are replicated locally.

Note Only Microsoft Internet Explorer 5.01 or later supports offline caching and replication.

[Show All](#)

About changing the design of a replicated database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Before you change the design of any objects in your [Design Master](#), follow these guidelines:

- [Synchronize](#) all [replicas](#) in the [replica set](#) before making design changes.
- Check all replicas to make sure that the name you are selecting for a new field or object isn't already being used as the name of a [local object](#). (If you do choose the same name, the local object will be renamed.)
- Don't create [relationships](#) between tables when you intend to [replicate](#) some tables but keep others local.
- Make sure that the number of bytes you add to a field or the number of fields you add to a record doesn't exceed the maximum size for a replicated database.
- If you use linked tables, keep in mind that linked tables aren't automatically replicated. If you create a linked table in the Design Master, it's created as a local object and remains local until you make it replicable. When you make the table replicable, the link and the table are sent to all the replicas in the set. After the table appears in each replica, you can change the link in an individual replica or in the Design Master without affecting the other members of the replica set. To refresh the link in all members of the replica set, change the replicable linked table to a local table (in the Design Master), refresh the link, and change the table back to a replicable table.
- Any formatting included in PivotTable and PivotChart views used in tables, queries, views, stored procedures, and functions is not replicated when you set up replication for your database. You can replicate the format of PivotTable and PivotChart views in a form.

[Show All](#)

About the replica set retention period setting (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

The [replica set](#) retention period setting controls the number of days non-[synchronized](#) records are retained in [system tables](#). The retention period is established when the database is initially made [replicable](#). If you replicate the database by using [Data Access Objects \(DAO\)](#) or [Jet and Replication Objects \(JRO\)](#), the default retention period is 60 days. If you replicate the database by using Microsoft Access or Briefcase, the default retention period is 1,000 days.

The retention period can be changed in a [Design Master](#) by setting the JRO **RetentionPeriod** property. The retention period must be between 5 and 32,000 days. The replica set should have a large retention period if the [replicas](#) do not synchronize frequently. However, if the replicas synchronize frequently and you want to keep the replica size small, specify a shorter retention period.

A replica or Design Master must receive, as well as send, data and schema within the retention period. When you open a replica that expires within 5 days in Microsoft Access, a message reminds you that the replica expires soon. If the retention period expires for a replica, Access will not synchronize changes between the expired replica and the other replicas in the replica set. If a replica does not synchronize with another replica in the set within the retention period, the next time you attempt to synchronize the replica, it gets removed from the replica set. If the Design Master expires, you must recover the Design Master to change an existing or new replica into the Design Master.

[Show All](#)

About AutoNumber field size and replicated databases (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

If you plan to [replicate](#) your database, you need to consider what field size is appropriate for the AutoNumber field used as a table's [primary key](#). If you use an AutoNumber field as the primary key for a table in the replicated database, set its **FieldSize** property to either **Long Integer** or **Replication ID**.

- If fewer than 100 records are routinely added between [synchronizing replicas](#), use a **Long Integer** setting for the **FieldSize** property to take up less disk space.
- If more than 100 records are routinely added between synchronizing replicas, you should use **Replication ID** for the **FieldSize** property setting to prevent records from being assigned the same primary key value in each replica. Note, however, that an AutoNumber field with a Replication ID field size produces a 128-bit value that will require more disk space.

[Show All](#)

Troubleshoot replication (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[I can't find the Design Master.](#)

The [Design Master](#) might have been moved from its original location or might be damaged. To find the new location:

1. Open a [replica](#) in the [replica set](#).
2. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**.
3. Click **Browse** to locate other members of the replica set on your computer or network.
4. Select a member of the replica set. If the **Make 'Filename' the Design Master** check box appears, the member you are [synchronizing](#) with is the Design Master.

[I can't open the Design Master.](#)

The [Design Master](#) might be damaged. Make a [replica](#) of the new Design Master.

[How?](#)

1. Make sure that neither the [replica](#) nor the current [Design Master](#) is in use elsewhere.
2. Open the replica that you want to make the new Design Master.
3. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**.
4. Select the current Design Master as the member to [synchronize](#) with.
5. Check the **Make 'Filename' the Design Master** check box.

[I want to recover the Design Master.](#)

You can recover a [Design Master](#) if the file has been lost, damaged, or moved. However, this option is unavailable if the currently opened [replica](#) is a [partial](#)

[replica](#).

1. Make sure that there is no Design Master in the [replica set](#).
2. Open the replica that you want to make the new Design Master.
3. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**.
4. Click a replica in the list, and then click **OK**.
5. Repeat steps 2 through 4 as many times as necessary to [synchronize](#) with each remaining replica in the replica set.

Important Make sure that you synchronize with all other replicas in the replica set.

6. On the **Tools** menu, point to **Replication**, and then click **Recover Design Master**.
7. Click **Yes**.

[I can't make my replica the Design Master.](#)

Your [replica](#) might be in use by another operation or application and can't be opened exclusively. Wait a few minutes, and try again.

If you still can't make your replica the [Design Master](#) and haven't received another error message, your replica may be damaged. Delete the replica and create a new one.

[There are two Design Masters in the replica set.](#)

Having more than one [Design Master](#) in a [replica set](#) prevents the members of the set from [synchronizing](#) correctly. If this occurs, do the following:

1. Determine which Design Master should continue to be the Design Master.
2. Delete the Design Master that should not exist.
3. Delete all [replicas](#) that had their most recent synchronization with the deleted Design Master or with other replicas that have had their most recent synchronization with the deleted Design Master.

[How?](#)

Caution

- Do not delete the [Design Master](#) for the [replica set](#) unless the Design Master is no longer functioning. Deleting the Design Master prevents you from changing the design of the database.
 - It's a good idea to check a [replica](#) for existing conflicts. Otherwise, the record of any existing conflicts will be lost when you delete (or move) the replica.
1. In Microsoft Windows Explorer, find the replica you want to delete.
 2. Click the replica that you want to delete and press the DELETE key.
 3. From another replica set member, attempt to [synchronize](#) with the deleted member. (This procedure broadcasts the deletion to the other members.)
4. Create new replicas from the remaining Design Master to replace the replica or replicas you deleted.

[When I change data in one member of the replica set, it doesn't change in another member.](#)

1. Open the member of the [replica set](#) that you want to [synchronize](#).
2. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**.
3. Enter the path and file name of the other [replicated](#) database with which you want the open database to synchronize.
4. Click **OK**.
5. Close and reopen the database to view the updated records.
6. If your data still doesn't appear, you may have changed data in [local objects](#), which are not synchronized with the members of the replica set, or there may have been a conflict with another member.

[When I change the design of an object in the Design Master, it doesn't change in any of the replicas.](#)

1. Open the [Design Master](#).
2. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**.

[I want to use a different program for resolving synchronization conflicts](#)

1. Open the [Design Master](#) for the [replica set](#).
2. On the **File** menu, click **Database Properties**.
3. Click the **Custom** tab.

4. In the **Name** box, type **ReplicationConflictFunction**.
5. In the **Type** box, click **Text**.
6. In the **Value** box, type the name of the custom function followed by **()**. For example, if the name of the custom function is MyResolver, type **MyResolver()**.

[I don't want to use replication when I drag a database file onto the My Briefcase icon.](#)

Briefcase [replicates Microsoft Access database](#) files that you drag into My Briefcase. If you don't want to replicate the database file, change the file extension from .mdb (or .mde) to another extension.

[I can't replicate my database created before Microsoft Access 97.](#)

Versions of Microsoft Access before Microsoft Access 95 can't [replicate](#) databases and likewise cannot use replicated databases.

You can convert a [replica set](#) that you created in Microsoft Access 97 by [synchronizing](#) your replica set and then converting every [replica](#) individually in Access 2000 or later versions.

[I don't want to replicate my forms, reports, macros, or modules when I replicate my database.](#)

Note This process must be performed before a database is [replicated](#).

1. On the **File** menu, click **Database Properties**.
2. Click the **Custom** tab.
3. Select the **ReplicateProject** property in the **Properties** box.
4. Click **No** next to **Value**.
5. Click **Modify**.

[I can't create a replica of my database while in Briefcase.](#)

The database file you selected isn't a [replica](#) .mdb file. To make a replica of your database, it must first be converted to a [Design Master](#). Converting your database increases the size of your file because new, hidden system fields and tables are added to your database.

[I can't change the design of my database while in Briefcase.](#)

If you want to be able to change the design of your database while working on the Briefcase copy of the .mdb file, make the Briefcase copy the [Design Master](#). It is important to remember, however, that no one will be able to make changes to the design of the [replica](#) remaining on the desktop.

[The formatting of my PivotTable and PivotChart views doesn't show up when I replicate my database](#)

Any formatting included in PivotTable and PivotChart views used in tables, queries, views, stored procedures, and functions is not replicated when you set up replication for your database. You can replicate the format of PivotTable and PivotChart views in a form.

[Show All](#)

Create a replicated database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Important

- It is recommended that you keep a backup copy of your original database. The backup copy can be used to create another [replica set](#) in emergency situations. However, note that any copy made from the backup copy of the database, or any [replica](#) made from a converted backup copy, will not be able to [synchronize](#) with the existing members in the original replica set.
- Before [replicating](#) your database, it is recommended that you read about the changes that are made to a database when it's replicated. For more information, see the Microsoft Access Help topic "Changes made to your database when you use replication."

Note Any formatting included in PivotTable and PivotChart views used in tables, queries, views, stored procedures, and functions is not replicated when you set up replication for your database. You can replicate the format of PivotTable and PivotChart views in a form.



[Create a replica of your database by using the menu bar](#)

1. Open the database you want to replicate. If you are working in a multiuser environment, make sure that all other users have closed the database.
2. If your database employs a database password to help protect it, remove the password.



[How?](#)

1. On the **File** menu, click **Open**.
2. Click the arrow to the right of the **Open** button, click **Open Exclusive**, and then open the database.
3. In the **Password Required** dialog box, type the database password, and then click **OK**. Passwords are [case-sensitive](#).

4. On the **Tools** menu, point to **Security**, and then click **Unset Database Password**. This command is only available if a database password was set previously.
5. In the **Unset Database Password** dialog box, type your current password.
3. On the **Tools** menu, point to **Replication**, and then click **Create Replica**.
4. Click **Yes** when you are asked whether you want to close the database.
5. Click either **Yes** or **No** (**Yes** is recommended) when you are asked whether you want to make a backup of the database (this dialog box is displayed only if the database must first be converted to a [Design Master](#) before the first replica is created).
6. In the **Location of New Replica** dialog box, do the following:

Select the location where you want to place the new replica.

Click the **Priority** button, type the priority for the replica, and click **OK**.

Check the **Prevent deletes** check box if you do not want users to delete records in the replica.

Select the replica [visibility](#) from the **Save as type** box.

Note When you replicate a database, all the tables, queries, forms, reports, macros, and modules are replicated by default, as well as the data.

[Create a replica with Briefcase](#)

1. You must have Briefcase installed on your computer. If Briefcase is installed, you will see a **My Briefcase** icon on the Microsoft Windows desktop. To install Briefcase on your computer in Windows 2000, right-click anywhere on your desktop, point to **New**, and click **Briefcase**.
2. If your database employs a database password to help protect it, remove the password.

[How?](#)

1. On the **File** menu, click **Open**.
2. Click the arrow to the right of the **Open** button, click **Open Exclusive**, and then open the database.

3. In the **Password Required** dialog box, type the database password, and then click **OK**. Passwords are [case-sensitive](#).
4. On the **Tools** menu, point to **Security**, and then click **Unset Database Password**. This command is only available if a database password was set previously.
5. In the **Unset Database Password** dialog box, type your current password.
3. In Windows Explorer, navigate to the [Microsoft Access database](#).
4. Drag the database file to the **My Briefcase** icon on the Windows desktop. You are given the option to create a backup copy of the original database.

[Show All](#)

Methods and properties used to programmatically replicate a database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can programmatically control the exchange and [synchronization](#) of information among members in a [replica set](#) in Microsoft Access 2000 and later versions by using [Jet and Replication Objects \(JRO\)](#). JRO can only be used with [Microsoft Access databases](#) (.mdb files).

For programmatically controlling [replication](#) with earlier databases (Microsoft Access 95 and Microsoft Access 97), you can use [Data Access Objects \(DAO\)](#).

[Replication with Jet and Replication Objects \(JRO\)](#)

You can create and maintain a replicated database by using Jet and Replication Objects (JRO) methods and properties.

Programming replicated databases

Use JRO with Microsoft Access databases (.mdb) if you require programmatic control over the exchange of data and design information among members of the replica set. For example, you can use JRO to write a procedure that automatically synchronizes a user's replica with the rest of the set when the user opens the database. The database must be closed to make the database replicable programmatically.

Methods

CreateReplica method

GetObjectReplicability method

MakeReplicable method

SetObjectReplicability method

Synchronize method

Properties

ActiveConnection property

ConflictFunction property

ConflictTables property

DesignMasterID property

ReplicaID property

ReplicaType property

RetentionPeriod property

TableName property

Visibility property

Methods and properties available in a partial replica

Append method

Delete method

PopulatePartial method

FilterCriteria property

FilterType property

Priority property



[Replication with Data Access Objects \(DAO\)](#)

You can create and maintain a replicated database in previous versions of Microsoft Access by using DAO methods and properties. Use DAO if you require programmatic control over the exchange of data and design information among members of the replica set. For example, you can use DAO to write a procedure that automatically synchronizes a user's replica with the rest of the set when the user opens the database.

You can use the following methods and properties to create and maintain a replicated database:

MakeReplica method

Synchronize method

ConflictTable property

DesignMasterID property

KeepLocal property

Replicable property

ReplicaID property

ReplicationConflictFunction property

Microsoft Jet provides these additional methods and properties for creating and maintaining partial replicas (replicas that contain a subset of the records in a full replica):

ReplicaFilter property

PartialReplica property

PopulatePartial method

[Show All](#)

Make a replicated database a regular database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

After you've converted a database into a [Design Master](#), you can't convert it back to its former status as a regular database. However, if you no longer want to use [replication](#) and want to decrease the size of a replicated database, you can create a new, regular database that contains all of the objects and data in your replicated database without the additional system fields, tables, and properties associated with replication.

Note For each table in the new database, create the same index used in the [replica](#) table. If the [s_GUID](#) is a [primary key](#) (and [foreign keys](#) in other tables refer to it), you'll want the s_GUID field in the new table. If the s_GUID field is not used as a primary key, there is no reason to preserve it in the new table. Don't copy the s_Lineage and s_Generation fields to the new table.

1. Create a blank database and open it.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. Click the replica that contains the database objects that you want in the regular database, and then click **Import**.
4. Click the database objects— except tables— that you want to import into the new database.
5. Click **OK** to import these objects.
6. In the old database, for each table in the replica, create a Make Table Query that takes all the data in the old table and creates a table in the new database with the same data.

 [How?](#)

1. Create a query, selecting the tables or queries that contain the records you want to put in the new table.

How?

1. In the [Database window](#), click **Queries**  under **Objects**, and then click **New** on the Database window toolbar.
2. In the **New Query** dialog box, click **Design View**, and then click **OK**.
3. In the **Show Table** dialog box, click the tab that lists the tables or queries whose data you want to work with.
4. Double-click the name of each object you want to add to the query, and then click **Close**.
5. Add fields to the **Field** row in the [design grid](#), and if you want, specify [criteria](#) and a sort order.
6. To view the query's results, click **View**  on the toolbar.
2. In query [Design view](#), click the arrow next to **Query Type**  on the [toolbar](#), and then click **Make Table**. The **Make Table** dialog box appears.
3. In the **Table Name** box, enter the name of the table you want to create or replace.
4. Do one of the following:

Click **Current Database**.

Click **Another Database**, and then either type the path of the database where you want to put the new table or click **Browse** to locate the database.

5. Click **OK**.
6. Drag from the [field list](#) to the query [design grid](#) the fields you want in the new table.
7. In the **Criteria** cell for the fields that you've dragged to the grid, type the [criteria](#).
8. To preview the new table before you create it, click **View**  on the toolbar. To return to query Design view and make changes or run the query, click **View**  on the toolbar again.
9. To create the new table, click **Run**  on the toolbar.

Note The data in the new table you create does not inherit the field properties or the [primary key](#) setting from the original table.

7. For each table in the new database, create the same [index](#) used in the replica's table.

[How?](#)

1. Open a table in [Design view](#).
 2. In the upper portion of the window, click the field that you want to create an [index](#) for.
 3. In the lower portion of the window, click in the **Indexed** property box, and then click **Yes (Duplicates OK)** or **Yes (No Duplicates)**.
8. For each table in the new database, create the [relationships](#) that existed for the replica's table.

[How?](#)

When you create a [relationship](#) between tables, the related fields don't have to have the same names. However, related fields must have the same [data type](#) unless the [primary key](#) field is an [AutoNumber](#) field. You can match an AutoNumber field with a Number field only if the **FieldSize** property of both of the matching fields is the same. For example, you can match an AutoNumber field and a Number field if the **FieldSize** property of both fields is **Long Integer**. Even when both matching fields are Number fields, they must have the same **FieldSize** property setting.

[Define a one-to-many or a one-to-one relationship](#)

1. Close any tables you have open. You can't create or modify relationships between open tables.
2. Press F11 to switch to the [Database window](#).
3. Click **Relationships**  on the toolbar.
4. If you haven't yet defined any relationships in your database, the **Show Table** dialog box is automatically displayed.

If you need to add the tables you want to relate and the **Show Table** dialog box isn't displayed, click **Show Table**  on the toolbar.

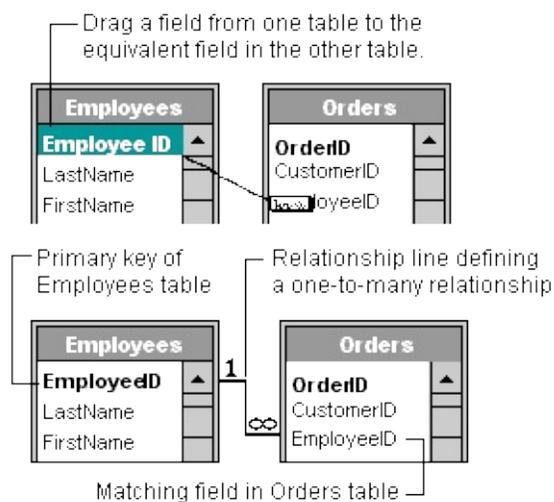
5. Double-click the names of the tables you want to relate, and then close the **Show Table** dialog box. To create a relationship between a table

and itself, add that table twice.

6. Drag the field that you want to relate from one table to the related field in the other table.

To drag multiple fields, press the CTRL key, click each field, and then drag them.

In most cases, you drag the primary key field (which is displayed in bold text) from one table to a similar field (often with the same name) called the [foreign key](#) in the other table.



7. The **Edit Relationships** dialog box is displayed. Check the field names displayed in the two columns to ensure they are correct. You can change them if necessary.

Set the relationship options if necessary.

8. Click the **Create** button to create the relationship.
9. Repeat steps 5 through 8 for each pair of tables you want to relate.

When you close the [Relationships window](#), Microsoft Access asks if you want to save the layout. Whether you save the layout or not, the relationships you create are saved in the database.

Note You can create relationships using queries as well as tables. However, [referential integrity](#) isn't enforced with queries.

[Define a many-to-many relationship](#)

1. Create the two tables that will have a [many-to-many relationship](#).
2. Create a third table, called a junction table, and add to the junction table new fields with the same definitions as the primary key fields from each of the other two tables. In the junction table, the primary key fields function as [foreign keys](#). You can add other fields to the junction table, just as you can to any other table.
3. In the junction table, set the primary key to include the primary key fields from the other two tables. For example, in an Order Details junction table, the primary key would be made up of the OrderID and ProductID fields.

[How?](#)

1. Open a table in [Design view](#).
2. Select the field or fields you want to define as the [primary key](#).

To select one field, click the [row selector](#) for the desired field.

To select multiple fields, hold down the CTRL key and then click the row selector for each field.

3. Click **Primary Key**  on the toolbar.

Note If you want the order of the fields in a multiple-field primary key to be different from the order of those fields in the table, click **Indexes**  on the toolbar to display the [Indexes window](#), and then reorder the field names for the [index](#) named PrimaryKey.

4. Define a [one-to-many relationship](#) between each of the two [primary tables](#) and the junction table.
9. Save your new database.

[Show All](#)

About replica visibility (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Replicas](#) fall into three visibility types: [global](#), [local](#), and [anonymous](#). A replica's visibility type determines several issues for the replica. For instance, a replica's visibility determines what type of replicas you can create from it, whether it can act as the [Design Master](#) in the [replica set](#), and how it handles conflicts during [synchronization](#). Visibility also determines which replicas that replica can synchronize with. You can't change a replica's visibility once you create the replica.

A global replica is the typical replica from which you can create all other types of replicas. Changes by a global replica are fully tracked and can be exchanged with any other global replica in the set. The global replica can also exchange changes with any local or anonymous replicas for which it becomes the [hub](#). The Design Master is a global replica. From a global replica, you can create replicas that are global, local, or anonymous. Replicas created from a global replica are global by default.

Local replicas synchronize only with their hub, a global replica. They are not permitted to synchronize with other replicas in the replica set. Other local replicas are not aware of the local replica. Only the hub replica is aware of local replicas, and only from it can you schedule an exchange with a local replica. All replicas created from a local replica will also be local and have the same parent replica.

An anonymous replica can synchronize with its parent, a global replica. Anonymous replicas subscribe by way of the Internet, and do not have any particular identity, but instead proxy their identity for updates to the publishing replica. Internet/intranet synchronization works well when the replica set remains small (fewer than 10 individual replicas) and the number of data inserts and updates are limited. Anonymous replicas provide a way of getting around this "limit on number of replicas" problem. In addition, using anonymous replicas helps to keep out unnecessary topology information about replicas that participate only occasionally. All replicas created from an anonymous replica

will also be anonymous and have the same parent replica. Anonymous replicas are recommended for use on the Internet for mass distribution because system-tracking information is not maintained, and replica size is reduced. A global replica will not be able to schedule synchronizations with an anonymous replica.

All local and anonymous replicas always have a priority of 0. Therefore, if any of their changes conflict with the global hub replica, the changes will automatically lose in any conflict resolution process. If they convey a non-conflicting change to the hub, the hub assumes authorship of the change.

[Show All](#)

A comparison of global, local, and anonymous replicas (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

There are three types of [replica visibility](#) that are available in Microsoft Access: [global](#), [local](#), and [anonymous](#). A replica's visibility type determines several issues for the replica. For instance, a replica's visibility determines what type of replicas it can create from itself, whether it can act as the [Design Master](#) in the [replica set](#), and how it handles conflicts during [synchronization](#). The visibility also determines which replicas that replica can synchronize with.

[Recommended replica use](#)

Global Use to serve as a [hub](#) in a replica set.
Local Use to control topology.
Anonymous Use to control topology/Internet.

[Recommended for a disconnected environment on the Internet](#)

Global Yes
Local Yes
Anonymous Yes. Anonymous replicas are recommended for Internet applications if you need a large number of replicas.

[Recommended for custom applications on the Internet](#)

Global Yes
Local Yes
 Yes. Use the Internet as a distribution mechanism for a large

Anonymous number of replicas.

[Use for replica creation](#)

Global Yes. Create a global, local, or anonymous replica.

Local Yes. Create new local replicas.

Anonymous Yes. Create new anonymous replicas.

[Use as a Design Master](#)

Global Yes. When a database is replicated, the first replica that is created (the Design Master) is a global replica.

Local No

Anonymous No

[Database replica visibility in Access 95 and 97](#)

Global Default.

Local Not applicable. Don't exist in these versions.

Anonymous Not applicable. Don't exist in these versions.

[Where conflict resolution is handled](#)

Global [Conflict](#) resolution can be managed at any global replica.

Local All conflicts are recorded and resolved at the [hub](#) (global) replica.

Anonymous All conflicts are recorded and resolved at the hub (global) replica.

[Where synchronization can occur](#)

Global Synchronizes with any global replica in the replica set, or any local or anonymous replica that the set created.

Local Synchronizes only with its [hub](#) (global) replica.

Anonymous Synchronizes only with its hub (global) replica.

[Who can schedule synchronization](#)

- Global Replication Manager can schedule synchronization with any global or local (child) replica. Synchronization cannot be scheduled with anonymous replicas.
- Local [Hub](#) replicas can schedule synchronization with a local replica. And local replicas can schedule synchronization with their hub (global) replica.
- Anonymous Anonymous replicas can schedule synchronization with their hub (global) replica.

[Briefcase support for synchronization](#)

- Global Yes
- Local No
- Anonymous No

[Show All](#)

About choosing a synchronization method (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Some of the content in this topic may not be applicable to some languages.

In Microsoft Access, Replication Manager, and [Jet and Replication Objects \(JRO\)](#), you can choose between three different methods to [synchronize](#) your data: direct, indirect and Internet.

[Direct synchronization](#)

[Direct synchronization](#) is an excellent solution when [replicas](#) are connected directly to the local area network and are available through shared network folders. Direct synchronization is not the best solution when remote synchronizing over a Remote Access Service (RAS) or dial-up connection is required. In this case you should use indirect or [Internet synchronization](#).

If you attempt to do a direct synchronization to a replica that's in the list of known replicas, and that replica isn't found, that replica will then be removed from the [replica set](#).

[Indirect synchronization](#)

[Indirect synchronization](#) is useful when you work in a disconnected environment, such as when you are traveling with a laptop.

Indirect synchronization can only be configured by using Replication Manager. Once indirect synchronization is configured with Replication Manager, you can execute an indirect synchronization by using Microsoft Access, Replication Manager, or JRO.

[Internet synchronization](#)

[Internet synchronization](#) is a convenient way to synchronize [replicas](#) in a disconnected environment in which an Internet Server is configured.

You must use Replication Manager to configure Internet synchronization.

[Show All](#)

About synchronization conflicts (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Conflicts can occur when data changes are made at two or more [replicas](#).



[Conflict behavior differences in different versions of Microsoft Access](#)

In Access 95 and 97, there were [synchronization](#) conflicts and synchronization errors. Synchronization conflicts occurred when two users updated the same record in two different databases in a [replica set](#). Synchronizing the two databases would succeed, but only one of the two sets of changes would be applied to both databases. Synchronization errors occurred when a change to data in one database in a replica set could not be applied to another database in the replica set because it would violate [referential integrity](#) or some other constraint.

In Access 2000 and later versions, the events that cause synchronization conflicts and errors are both viewed simply as synchronization conflicts. A single mechanism is used to record and resolve the conflicts, making the process much easier. Whenever a conflict occurs, a winning change is selected and applied in all replicas, and the losing change is recorded as a conflict at all replicas. The **Conflict Viewer**, the default tool in Access, is used to reconcile and resolve synchronization conflicts.



[A comparison of row and column level conflict resolution between different versions of Access](#)

In Microsoft Access 95 and 97, conflicts are determined at the row-level. Conflicts will occur whenever the same record is changed in two separate replicas, even if different fields are changed at each replica. For example, changing a customer's phone number in one replica and that same customer's zip code in another replica will cause a [conflict](#).

In Access 2000 and later versions, conflicts are determined at the column-level. Changes to the same record in two different replicas cause a [synchronization](#) conflict only if the same column, or field, is changed. So in the previous example, there would no longer be a synchronization conflict since the two users changed different fields.

Column-level conflict resolution is the default when a database is made replicable. To specify row-level conflict resolution for a table, you must set it prior to making the table replicable.

You can set the conflict tracking on a database or an individual table prior to making it replicable. Once you create a replica, you can't change the setting. However, you can change the value of individual tables to row-level tracking.

Note By using [Jet and Replication Objects \(JRO\)](#) you can only specify the default tracking for the database, not for individual tables. When making a database replicable, you can set **ColumnLevelTracking** to either **True** or **False** by using the JRO **MakeReplicable** method.

[Types of synchronization conflicts](#)

Simultaneous update conflict

A simultaneous update conflict occurs when two replicas both update the same record. The losing record is logged in the conflict table.

Unique key conflict

A unique key conflict can occur in one of the following ways:

- Two replicas both enter a new record with the same value in a field that is a [primary key](#) or has an [unique index](#).
- The [Design Master](#) creates a unique index, and a replica simultaneously adds two or more records with the same key value. When the schema change reaches the first replica with duplicate key records, then the first duplicate key record remains in the base table. Subsequent records are marked for deletion and are written to the conflict table.

Table-level validation conflict

A table-level validation conflict occurs when data is entered that breaks a table-level [validation rule](#). A validation rule restricts the values or types of data that can be entered into a table. If you add a table-level validation rule to the Design Master without determining if any existing data violates the rule, you may encounter a conflict when you [synchronize](#) the design changes to the rest of the [replica set](#). In this case, the records are deleted and logged in the conflict table.

Update referential integrity conflict

An update [referential integrity](#) conflict occurs when the primary key is updated at one replica and new child records that reference the original primary key value are added at a different replica. During synchronization, the new child records are deleted and logged in the conflict table. To reduce the number of these types of errors during synchronization, you may want to use the s_GUID field as the primary key in some or all of your replicated tables.

Delete referential integrity conflict

A delete referential integrity conflict occurs when a primary key record is deleted in one replica while new child records that reference the deleted primary key are added at a second replica. When the two replicas are synchronized, the new child records are marked for deletion and added to the conflict table.

Locking conflict

A locking conflict occurs when the record cannot be applied during synchronization because another user locked the table. Microsoft Access will try to update the record several times, but if unsuccessful, the synchronization will stop, and the entire transaction will return to its original status. An error is returned but no conflict is logged.

Foreign key violation conflict

A [foreign key](#) violation conflict occurs when there is an invalid primary key record. This could be caused by any of the other conflict types.

Case-sensitivity and sort order conflicts

The number of conflicts can increase in replica sets that reside across multiple database types— Access and Microsoft SQL Server for instance— if language sort orders or the case-sensitivity of the sorting is different. This is because unique key values in one database may not be unique in the database with different sorting. The problem can affect both indexed text data and metadata, but it is not an issue for non-indexed text data. Creating replicas with different sorting will require special care to ensure that metadata and indexed text will always be unique for both methods of sorting.

Conflicts in partial replicas

A [partial replica](#) receives conflicts associated with all of the rows in the partial replica, even for those rows that are added to the partial replica during synchronization.

[Resolving conflicts](#)

If the same field in a replicated database is changed in one or more replicas, conflicts result when you [synchronize](#) a replica with the [replica set](#). To resolve these conflicts, open the **Conflict Viewer**. The wizard presents each conflict, and you must manually determine which changed record contains the correct information.

[Show All](#)

Synchronize and resolve conflicts in a replica set (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Synchronization is the process of updating two members of a [replica set](#) by exchanging all updated records and objects in each member. Two replica set members are synchronized when the changes in each have been applied to the other. Conflicts can occur if data has changed in the same field of two replica set members that have been [synchronized](#).



[Synchronize with another replica set member in a database](#)

1. Open the replica set member you want to synchronize.
2. On the **Tools** menu, point to **Replication**, and then click **Synchronize Now**. (If a Synchronizer on the Internet manages any replica set member, you will see that the managed [replica](#) appears in the **Directly with Replica** drop-down list box as `http://ServerName`, where *ServerName* is the name of the Internet Server.
3. Enter the path and file name of the replica set member you want to synchronize with the current database.
4. If your database is the [Design Master](#) and you want to make the other replica the Design Master, select the **Make 'Filename' the Design Master** check box. (This option is unavailable if either of the replicas to be synchronized is a [partial replica](#).)
5. Click **OK**.
6. When asked whether you want to close and reopen the database, click **Yes**. (If you click **No**, you might not see all of the changes.)



[Synchronize a replica by using Briefcase](#)

1. Double-click the **My Briefcase** icon, and then click the database file.
2. On the **Briefcase** menu, click **Update Selection**.
3. Click **Update** to have Briefcase exchange the updates.

Note Briefcase does not support [local](#) or [anonymous replicas](#).

[Resolve synchronization conflicts between replica set members](#)

1. Start the Microsoft Replication **Conflict Viewer** by using one of the following steps:

Open the database that you want to check for conflicts. If there are conflicts, you are asked whether you want to resolve them. Click **Yes** to proceed.

On the **Tools** menu, point to **Replication**, and then click **Resolve Conflicts**.

2. In the **Conflict Viewer**, select a table that has conflicts from the list, and click **View**.
3. For each conflict shown in the **Conflict Viewer**, choose one of the following options.

To keep the data in Replica 1, click **Keep existing data**.

To modify the data in Replica 1 and copy it to Replica 2 on the next synchronization, click **Keep revised data**.

To keep the data in Replica 2 and overwrite the data in Replica 1 on the next synchronization, click **Overwrite with conflicting data**.

To modify the data in Replica 2 and overwrite the data in Replica 1 on the next synchronization, click **Overwrite with revised data**.

4. Click **Resolve**.
5. Repeat steps 3 and 4 as many times as necessary to resolve each conflict in the table.

Note The record of conflicts can be lost if you delete or move a [replica](#). Before deleting or moving a replica, check it for conflicts.

[Use a different program for resolving synchronization conflicts](#)

1. Open the [Design Master](#) for the [replica set](#).
2. On the **File** menu, click **Database Properties**.

3. Click the **Custom** tab.
4. In the **Name** box, type **ReplicationConflictFunction**.
5. In the **Type** box, click **Text**.
6. In the **Value** box, type the name of the custom function followed by (). For example, if the name of the custom function is MyResolver, type **MyResolver()**.

[Show All](#)

About setting replica priority (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[Replica](#) priority affects how changes to records are resolved when there is a [synchronization](#) conflict.



[Priority differences in different versions of Access](#)

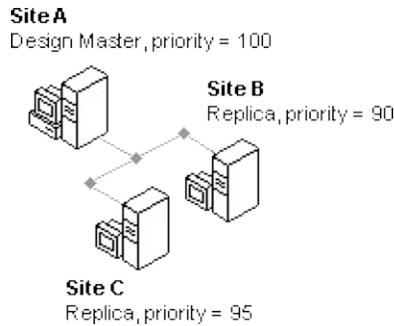
In Microsoft Access 2000 and later versions, a replica in a [replica set](#) is assigned a priority when it is created, and the highest priority replica wins in the case of a synchronization conflict. Where priorities are equal, the replica with the lowest **ReplicaID** property wins. In Microsoft Access 95 and 97, synchronization conflicts are resolved by using the rule in which the winning record is the one changed most often.



[Determine replica priority](#)

Every replica is assigned a priority number between 0 and 100, with 100 as the highest priority. When a database is made replicable, the replica's default priority is set to 90. Subsequent replicas have a default priority equal to 90 percent of the [hub](#)'s priority. Priorities for [local](#) and [anonymous replicas](#) are always 0. Local and anonymous replicas automatically lose if their changes conflict with their [global](#) hub replica. If a local or anonymous replica sends a non-conflicting change to the hub, the hub assumes ownership of the change.

Microsoft Access carries the historical priority for each record in a [replica set](#). During synchronization, the historical priority setting is evaluated and the record with the highest priority wins in any conflicts. To understand the priority scheme, consider the example of one [Design Master](#), Replica A, with a priority of 100. Replica A replicates to two replicas, Replica B, which has a priority of 90, and Replica C, which has a priority of 95.



In a simple case, the highest priority change wins. Suppose that all three sites agree initially that Replica A created version one of the record, and no subsequent updates have occurred. If Replicas A and B update the record simultaneously, then Replica A's update is the conflict winner because it has the highest priority.

If multiple changes occurred to the same record after the last synchronization, then the replica that has changes with the highest historical priority is used to determine the conflict winner. For example, suppose that Replica A makes version two, and then sends it on to Replica B. Replica B makes version three, and then sends it back to Replica A, where it is merged. Then suppose that Replica C has also made a version two, and reconciles with Replica A. Choosing the highest historical priority of changes that occurred after the changes to the original record (Replica A with a replica priority 100 and 95 for the version of the row that Replica C has), Replica A and B's joint changes are the priority winner when compared to the change made by Replica C. However, because Replica A was created before Replica B, Replica A is the conflict winner. One benefit of this decision rule is that no changes made at the highest priority replica ever get reversed by the conflict resolution mechanism.

[Show All](#)

Overview of Access security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

There are different strategies for controlling the level of access to your [Microsoft Access database](#) and its objects. These strategies are briefly described below and are roughly arranged by level of security.

[Encoding/decoding](#)

The simplest method of protection is to encode the database. Encoding a database compacts the database file and helps protect it from being read by a word processor. Encoding a database that employs no security measures has no effect because anybody can open the database and gain full access to all objects in the database. Encoding is particularly useful when you transmit a database electronically, or when you store it on floppy disk, tape, or compact disc.

Before you can encode or decode a Microsoft Access database, you must be either the [owner](#) of the database or, if the database employs security measures, a member of the [Admins group](#) of the [workgroup information file](#) that contains the accounts used to help protect the database. You must also be able to open the database in [exclusive](#) mode, which means you must have Open/Run and Open Exclusive [permissions](#).

Decoding a database reverses the encoding.

[Show or hide objects in the Database window](#)

Another method of protecting the objects in your database from casual users is to hide the objects in the Database window. This method of protection provides the least security because it is relatively simple to show any hidden objects.

[Using startup options](#)

You use the startup options to specify settings such as a startup form, which

opens automatically when your database opens, and your database application title and icon. You can also hide the Database window, and set your own switchboard form. In a new database, the startup properties do not exist until a user makes a change to the default settings in the **Startup** dialog box.

[With a password](#)

Another simple method of providing security is to set a password for opening the Microsoft Access database. Once a password is set, a dialog box that requests the password will be displayed whenever the database is opened. Only users who type the correct password will be allowed to open the database. Once a database is open, all its objects are available to the user (unless other types of security have already been defined, as described later in this topic). For a database that is shared among a small group of users or on a single computer, setting a password is often all that is required.

Microsoft Access stores the database password in an unencrypted form. If this will compromise the security of the password-protected database, you should not use a database password to protect the database. Instead, you should define user-level security to help control access to sensitive data in that database.

Do not use a database password if you will be [replicating](#) a database. Replicated databases can't be [synchronized](#) if database passwords are defined.

[With user-level security](#)

The most flexible and extensive method of implementing security measures for a database is called [user-level security](#). With user-level security, you can establish different levels of access to sensitive data and objects in your database. To use a database with user-level security, users type a password when they start Microsoft Access. Access reads a [workgroup information file](#), where each user is identified by a unique identification code. The level of access and the objects that a user has access to are established based on this identification code and password.

Although setting up user-level security on most databases can be a daunting task, the User-Level Security Wizard makes it easy to quickly help protect your Access database in a one-step process. Furthermore, by implementing common security schemes, the User-Level Security Wizard minimizes and may even

eliminate the need to use the **Security** command from the **Tools** menu.

After running the User-Level Security Wizard, you can create your own groups of users and assign or remove [permissions](#) for various users or group of users for a database and its existing tables, queries, forms, reports, and macros. You can also set the default permissions that Microsoft Access assigns for any new tables, queries, forms, reports, and macros that are created in a database. Permissions are granted to groups and users to regulate how they are allowed to work with each table, query, form, report, and macro in a database.

 [Preventing users from replicating a database, setting passwords, or setting startup options](#)

In a multiuser environment, there are many situations where you might need to employ security measures for your database. You might want to prevent users from [replicating](#) a database. Replicating a database allows a user to make a copy of a shared database and also adds fields and makes other changes to the current database. You might want to prevent users from setting a database password, because if they do, no other user will be able open the database without providing that password. You might also want to keep users from changing startup properties that specify features such as custom menus, custom [toolbars](#), or the startup form.

If a shared database doesn't have [user-level security](#) defined, you can't prevent a user from making any of these changes. When user-level security is defined, a user or group must have Administer [permission](#) for the database to replicate a database, set a database password, or change its startup properties. Only members of the [Admins group](#) of the current [workgroup](#) have Administer permission.

If a user or group currently has Administer permission for a database, removing that permission will prevent the user or group from making any of these changes. If you need to allow a user or group to perform any of these tasks, you can assign the Administer permission to that user or group. You can't control access to these three tasks independently.

 [Securing Microsoft Visual Basic for Applications \(VBA\) Code](#)

There are two different ways you can help protect [Microsoft Visual Basic for](#)

[Applications \(VBA\)](#) code in standard [modules](#) and [class modules](#) (such as code behind forms and reports).

You can help protect your code with a password, which you enter once per session. The password helps prevent unauthorized users from editing, cutting, pasting, copying, exporting, and deleting VBA code.

When your VBA code is error-free and working correctly, you can remove it from your database by saving it as an MDE file. This helps protect the intellectual property of your code, and prevent modifications to the design of forms, reports, and modules in your database.



[Securing data access pages](#)

[Data access pages](#) are [HTML](#) pages that contain references to the data in a [Microsoft Access file](#). However, the data access pages aren't actually stored in the Access file; they are stored as HTML files either in the local file system, in a folder on a network share, or on an [HTTP](#) server. For this reason, Microsoft Access has no control over the security of data access page files. To help protect the data access page, you must employ security measures for its link and HTML file by using the file system security of the computer where you have stored these files. To help protect the data accessed by the page, you must either employ security measures for the database that the page is connected to or configure Microsoft Internet Explorer security settings to prevent unauthorized access.

Note Before you distribute the database or data access page, you should remove any personal information that may be stored in the file. See [Remove personal information from an Access file or page](#) for more information.

[Show All](#)

Troubleshoot security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[I can't encode \(or decode\) my database.](#)

You must either be the [owner](#) of the database or a member of the [Admins group](#) with Open Exclusive [permission](#) to encode or decode a database employing [user-level security](#).

Also, you must have enough storage space on your disk for both the original and the encoded or decoded versions of the database.

[I can't set a database password.](#)

You can't set a database password if [user-level security](#) has been defined for your database and you don't have Administer [permission](#) for the database.

[I set a database password and am having problems with some objects in my database.](#)

A database password is defined in addition to [user-level security](#). If user-level security has been defined, any restrictions based on user-level security [permissions](#) remain in effect.

[I set a password on my replicated database and now I can't synchronize it with other databases.](#)

Do not use a database password if you will be [replicating](#) a database. Replicated databases can't be [synchronized](#) if database passwords are defined.

[I want to restore a damaged or deleted Microsoft Access workgroup information file.](#)

In rare circumstances, the Microsoft Access [workgroup information file](#) can

become damaged. If you start Microsoft Access and that file can't be opened, a message will be displayed. How you solve this problem depends on whether you have a backup copy and how your workgroup information file was originally created or specified. The following table summarizes how you should address each situation.

Important Microsoft Access stores security account information in the workgroup information file. So in all cases where you don't have a backup copy and have to re-create a workgroup information file, if you've restricted [permissions](#) to your databases, you must re-create security accounts with the same [case-sensitive](#) names and [personal ID \(PID\)](#) entries as before.

Note You don't have to redefine permissions or object ownership because this information is stored in the security-enabled databases.

Used Workgroup Administrator	Made backup copy	Solution
No. Used default file created when installing Microsoft Access.	No	Re-start Microsoft Access and Microsoft Windows installer will re-create a default workgroup information file.
No. Used default file created when installing Microsoft Access.	Yes	Use Windows Explorer, My Computer, MS-DOS copy command, or backup software to copy the most recent copy of the file to the folder where you installed Microsoft Access.
Yes. Created a new file.	No	Run the Workgroup Administrator again, typing the same case-sensitive name, organization, and workgroup ID (WID) entries you used when you originally created it.
Yes. Created a new file.	Yes	Copy or restore the backup copy to the path where you originally saved your workgroup information file.
Yes. Joined (specified) a file on a path other than the folder	Yes	Copy or restore the backup copy to the original path.

where Microsoft Access is installed.

Yes. Joined (specified) a file on a path other than the folder No where Microsoft Access is installed.

Create a new copy by using the method that was used to create the original file: re-start Microsoft Access so that installer will re-create a default workgroup information file, or run the Workgroup Administrator and type the same case-sensitive name, organization, and workgroup ID entries used when it was originally created.

[I want to rebuild a workgroup information file from user and group names and IDs.](#)

If a [workgroup information file](#) becomes damaged or has been deleted, and a backup copy isn't available, you can re-create the workgroup information file if you have the exact [case-sensitive](#) information that you used to create the file and define the accounts and groups in the file originally.

1. Create a new workgroup information file, making sure to enter the exact case-sensitive name, company name, and [workgroup ID \(WID\)](#) that you used to create the original file. Failure to re-enter the exact entries used to create the original file will create an invalid [Admins group](#).

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also know as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

2. Re-create any [group accounts](#), making sure to enter the exact case-sensitive group name and [personal ID \(PID\)](#) for each group.

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup

before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

3. Re-create each [user account](#), making sure to enter the exact case-sensitive user name and PID for each user.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
 - The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).
4. If you saved the new workgroup information file to a new name or location in step 1, tell other users in the [workgroup](#) to use the Workgroup Administrator to join the new workgroup information file.

[Show All](#)

About making connections to external data sources more secure

When you define a connection to an external data source for a PivotTable list on a Web page, such as a data access page, the authoring environment stores this information as an unencrypted (plain text) connection string in the HTML page. As a result, a user opening the page in a browser can easily view the HTML source for the page and read the connection string, which may include a user name and password.

To prevent unauthorized access using information from the connection string, if the data source supports this mode of authentication, use Windows Authentication (also referred to as a Trusted Connection and Integrated Security), which uses the current user's Microsoft Windows® account to connect to an external data source. Using Windows Authentication to connect to an external data source has the following advantages:

- The page designer doesn't have to enter a user name or password to connect to the data source, so that information cannot be exposed in the page's HTML source.
- The user of the page doesn't have to enter a user name or password to connect to the data source when opening the page, so that information cannot be compromised while it's being sent to the server.
- Only a Windows user account that is configured in the security system for the data source will be allowed to connect to that data source.

For example to connect to Microsoft SQL Server using Windows Authentication, use the following procedure when defining a connection.

Important Before you can use Windows Authentication to connect to a Microsoft SQL Server database, a server administrator must configure the server to use this mode of authentication, and must grant your Windows user account (or a group of which your user account is a member) login access. Additionally, the server administrator should provide your account with the minimum level of permissions required for your PivotTable list.

1. In [Design view](#) of a data access page, click **Toolbox**  on the toolbar.
2. In the [toolbox](#), click the **Office PivotTable** tool .
3. Click inside the PivotTable list to activate it.
4. Click **Commands and Options** on the toolbar, and then click the **Data Source** tab.
5. Click **Edit**.
6. In the **Data Link Properties** dialog box, double-click **Microsoft OLE DB Provider for SQL Server** on the **Provider** tab.
7. On the **Connection** tab, specify the name of the server.
8. Select **Use Windows Integrated security**.
9. Select a database and click **OK**.

If Windows Authentication is not available for the data source you want to connect to, you must enter a user name and password to connect to the data source. Because this user name and password might be viewed when the PivotTable list is on a Web page, do not use this mode of authentication to connect to sensitive data from a PivotTable list on a Web page. Even if you are not concerned about unauthorized users viewing your data, you should connect only with a user account that has limited permissions on the data source. For example, when connecting to a SQL Server database using a user name and password, do not use the SA account or any other account that has elevated permissions, because an unauthorized user might be able to use this account and password to access other data on the server.

Note If you use the **Data Retrieval Service for Microsoft SQL Server** (or use an **Data Retrieval Service Connections (.udc)** file that is defined to use that data retrieval service) to connect to an external data source, and you do not use Windows Authentication, the user name and password used to connect to the data source are not saved in the Web page. When you use this method to connect to an external data source, users of the Web Page that hosts your PivotTable list will be prompted to enter their login information.

[Show All](#)

About distributing a security-enhanced application (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The security of an application depends on users having access to both an application database where [permissions](#) are stored, and a [workgroup](#) where [user](#) and [group accounts](#) and passwords are stored. Depending on how the application is to be used, you can make the application and workgroup available to users in a variety of ways.

[Copy the application files and the workgroup information file that defines the workgroup to a network server location](#)

Using the Workgroup Administrator, users can join the workgroup by specifying the network server path to the [workgroup information file](#), and then run the application directly from the server location. You might want to prevent users from opening the application with exclusive access by removing Open Exclusive permissions for the database from the [Users group](#).

[Give each user a copy of the application files and the workgroup information file that defines the workgroup](#)

Using the Workgroup Administrator, users can join the workgroup by specifying the path to the [workgroup information file](#) on their computer, and then running their copies of the application on their computers. However, if you have to update the application, its data, or the workgroup information file, you'll have to give each user copies of all updated files.

As an alternative, you might want to split the application into a data database placed on a network server and an application database containing all other database objects placed on each user's computer— a database configuration typically known as a [front-end/back-end application](#). Typically, in this configuration, you want to keep the workgroup information file on the server as

well. Then if you have to update the application, you only have to give users updated versions of the application database file.

[Give users in different Microsoft Access workgroups access to the application by creating the same group in each workgroup](#)

You can create a group account in each workgroup with an identical name and [personal ID \(PID\)](#), and then add users in each workgroup to that group. For example, a user working on a network on a Microsoft Access workgroup in Honolulu can use an application created for a Microsoft Access workgroup in Melbourne if that user is a member of a group account that was created with an identical name and PID on both workgroups.

[Show All](#)

About user-level security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Microsoft Access [user-level security](#) is very similar to the security mechanisms seen on server-based systems. By using passwords and [permissions](#), you can allow or restrict the access of individuals, or groups of individuals, to the objects in your database. Security accounts define the users and groups of users allowed access to the objects in your database. This information, known as a [workgroup](#), is stored in a [workgroup information file](#).

[About user-level security](#)

The best method of helping secure a database is called user-level security. The two main reasons to use user-level security are to:

- Prevent users from inadvertently breaking an application by changing tables, queries, forms, reports, and macros on which the application depends.
- Help protect sensitive data in the database.

Under user-level security, users type a password when they start Microsoft Access. Access then goes out and reads a workgroup information file, where each user is identified by a unique identification code. Within the workgroup information file, users are identified as authorized individual users, and as members of specific groups, by their personal ID and password. Microsoft Access provides two default groups: administrators (named the [Admins group](#)) and users (named the [Users group](#)), but you can define additional groups.

Although setting up user-level security on most databases can be a daunting task, the User-Level Security Wizard makes it easy to employ innovative security features for your [Access database](#) in a one-step process. Furthermore, by implementing common security schemes, the User-Level Security Wizard may even eliminate the need to use the **Security** command on the **Tools** menu.

The User-Level Security Wizard will help you to assign permissions and create

[user](#) and [group accounts](#). However, after running the wizard, you can manually assign, modify, or remove permissions for user and group accounts in your workgroup for a database and its existing tables, queries, forms, reports, and macros. You can also set the default permissions that Microsoft Access assigns for any new tables, queries, forms, reports, and macros that are created in a database.

Permissions are granted to groups and users to regulate how they are allowed to work with each table, query, form, report, and macro in a database. For example, members of the Users group might be allowed to view, enter, or modify data in a Customers table but not to change the design of that table. Members of the Users group might be allowed to only view data in a table containing order data but be totally denied any access to a Payroll table. Members of the Admins group have full permissions on all of a database's tables, queries, forms, reports, and macros. You can have more detailed control by creating your own group accounts, assigning appropriate permissions to those groups, and then adding users to those groups.

If you only need an administrators group and users group for your security purposes, you don't need to create additional groups; you can use the default Admins and Users groups. In this case, you only need to assign the appropriate permissions to the default Users group, and add any additional administrators to the default Admins group. Any new users you add are automatically added to the Users group. Typical permissions for the Users group might include Read Data and Update Data for tables and queries, and Open/Run for forms and reports.

If you need more detailed control of different groups of users, you can create your own groups, assign different sets of permissions to those groups, and then add users to the appropriate groups. To simplify the management of permissions, it is recommended that you only assign permissions to groups (not users), and then add users to the appropriate groups.

For example, you could help protect an Orders database by creating a Managers group for managers, a Sales Reps group for sales representatives, and a Staff group for staff employees. You can assign the least restrictive set of permissions to the Managers group, a more restrictive set of permissions to the Sales Reps group, and the most restrictive set of permissions to the Staff group. When you create a user account for a new employee, you add that account to the appropriate group. The employee then has the permissions associated with that

group.

[About workgroups and workgroup information files](#)

A Microsoft Access workgroup is a group of users in a multiuser environment who share data. If user-level security is defined, the members of a workgroup are recorded in [user](#) and [group accounts](#) that are stored in a Microsoft Access workgroup information file. Users' passwords are also stored in the workgroup information file. These security accounts can then be assigned permissions for databases and their tables, queries, forms, reports, and macros. The permissions themselves are stored in the security-enabled database.

The first time a user runs Microsoft Access after Microsoft Office setup, Access will automatically create a Microsoft Access workgroup information file that is identified by the name and organization information that the user specifies. The relative location of this workgroup information file is then added to the following registry keys:

HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\Access\Jet\4.0\Engin

and

HKEY_USERS\.DEFAULT\Software\Microsoft\Office\11.0\Access\Jet\4.0\Engin

Subsequent users will inherit the default workgroup file path from the HKEY_USERS registry key. Because this information is often easy to determine, it's possible for unauthorized users to create another version of this workgroup information file and consequently assume the irrevocable [permissions](#) of an administrator account (a member of the [Admins group](#)) in the workgroup defined by that workgroup information file. To prevent this, create a new workgroup information file, and specify a [workgroup ID \(WID\)](#). Only someone who knows the WID will be able to create a copy of the workgroup information file.

Any user and group accounts or passwords that you create are saved in this workgroup information file unless the user joins a different workgroup using the Workgroup Administrator.

Note In Access 2002 or later, the Workgroup Administrator can now be run

from the **Security** submenu on the **Tools** menu.

Be sure to write down your exact name, organization, and workgroup ID — including whether letters are uppercase or lowercase (for all three entries) — and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.



[About how permissions work and who can assign them](#)

There are two types of [permissions](#): explicit and implicit. Explicit permissions are those permissions that are granted directly to a [user account](#); no other users are affected. Implicit permissions are those permissions that are granted to a [group account](#). Adding a user to that group grants the group's permissions to that user; removing a user from the group takes away the group's permissions from that user.

When a user attempts to perform an operation on a database object that employs security features, that user's set of permissions are based on the intersection of that user's explicit and implicit permissions. A user's security level is always the least restrictive of that user's explicit permissions and the permissions of any and all groups to which that user belongs. For this reason, the least complicated way to administer a workgroup is to create new groups and assign permissions to the groups, rather than to individual users. Then you can change individual users' permissions by adding or removing them from groups. Also, if you need to grant new permissions, you can grant them to all members of a group in a single operation.

Permissions can be changed for a database object by:

- Members of the [Admins group](#) of the workgroup information file in use when the database was created.
- The [owner](#) of the object.
- Any user who has Administer permission for the object.

Even though users might not be able to currently perform an action, they might be able to grant themselves permissions to perform the action. This is true if a user is a member of the Admins group, or if a user is the owner of an object.

The user who creates a table, query, form, report, or macro is the owner of that object. Additionally, the group of users that can change permissions in the database can also change the ownership of these objects, or they can re-create these objects, which is another way to change ownership of the objects. To re-create an object, you can make a copy of the object, or import it or export it to another database. This is the easiest way to transfer the ownership of objects, including the database itself.

Note Copying, importing, or exporting doesn't change the ownership of a query that has its **RunPermissions** property set to **Owner's**. You can change ownership of a query only if its **RunPermissions** property is set to **User's**.

 [About organizing security accounts](#)

A Microsoft Access workgroup information file contains the following predefined accounts.

Account	Function
Admin	The default user account . This account is exactly the same for every copy of Microsoft Access and other applications that can use the Microsoft Jet database engine , such as Microsoft Visual Basic for Applications (VBA) and Microsoft Excel.
Admins	The administrator's group account . This account is unique to each workgroup information file. By default, the Admin user is in the Admins group . There must be at least one user in the Admins group at all times.
Users	The group account comprising all user accounts. Microsoft Access automatically adds user accounts to the Users group when a member of the Admins group creates them. This account is the same for any workgroup information file, but it contains only user accounts created by members of the Admins group of that workgroup. By default, this account has full permissions on all newly-created objects. The only way to remove a user account from the Users group is for a member of the Admins group to delete that user.

In effect, security in Microsoft Access is always active. Until you activate the logon procedure for a workgroup, Microsoft Access invisibly logs on all users at startup by using the default [Admin user account](#) with a blank password. Behind

the scenes, Microsoft Access uses the Admin account as the administrator account for the workgroup, as well as the [owner](#) of any databases and tables, queries, forms, reports, and macros created.

Administrators and owners are important because they have permissions that can't be taken away:

- Administrators (members of the Admins group) can always get full permissions for objects created in the workgroup.
- An account that owns a table, query, form, report, or macro can always get full permissions for that object.
- An account that owns a database can always open the database.

Because the Admin user account is exactly the same for every copy of Microsoft Access, the first steps in securing your database are to define administrator and owner user accounts (or use a single user account as both the administrator and owner account), and then to remove the Admin user account from the Admins group. Otherwise, anyone with a copy of Microsoft Access can log on to your workgroup by using the Admin account and have full permissions for the workgroup's tables, queries, forms, reports, and macros.

You can assign as many user accounts as you want to the Admins group, but only one user account can own the database itself—the user account that is active when the database is created, or when ownership is transferred by creating a new database and importing all of a database's objects into it. However, group accounts can own tables, queries, forms, reports, and macros within a database.

[Considerations when organizing security accounts:](#)

- Only user accounts can log on to Microsoft Access; you can't log on by using a group account.
- The accounts that you create for users of the database must be stored in the workgroup information file that those users will join when they use the database. If you're using a different file to create the database, change the file before creating the accounts.
- Make sure to create a unique password for your administrator and user accounts. A user who can log on by using the administrator account can always get full permissions for any tables, queries, forms, reports, and

macros created in the workgroup. A user who can log on using an owner account can always get full permissions for those objects owned by that user.

After you create user and group accounts, you can view and print the relationships between them. Microsoft Access prints a report of the accounts in the workgroup, showing the groups to which each user belongs and the users that belong to each group.

Note If you are using a workgroup information file created with Microsoft Access version 2.0, you must be logged on as a member of the Admins group to print user and group information. If the workgroup information file was created with Microsoft Access version 7.0 or later, all users in the workgroup can print user and group information.

[Show All](#)

Remove user-level security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

1. Start Microsoft Access.
2. Open the database that employs [user-level security](#).
3. Log on as a workgroup administrator (a member of the [Admins group](#)).
4. Give the [Users group](#) full [permissions](#) on all tables, queries, forms, reports, and macros in the database.

 [How?](#)

1. Open the database.

The [workgroup information file](#) in use when you log on must contain the [user](#) or [group accounts](#) that you want to assign permissions for at this time; however, you can assign [permissions](#) to groups and add users to those groups later.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Permissions**.
3. On the **Permissions** tab, click **Users** or **Groups**, and then in the **User/Group Name** box, click the user or group that you want to assign permissions to.
4. Click the type of object in the **Object Type** box, and then click the name of the object to assign permissions for in the **Object Name** box. Select multiple objects in the **Object Name** box by dragging through the objects you want to select, or by holding down CTRL and clicking the objects you want.

Note Hidden objects aren't displayed in the **Object Name** box unless you select **Hidden objects** on the **View** tab of the **Options** dialog box (**Tools** menu).

5. Under **Permissions**, select the permissions you want to assign, or clear the permissions you want to remove for the group or user, and then

click **Apply**. Repeat steps 4 and 5 to assign or remove permissions for additional objects for the current user or group.

6. Repeat steps 3 through 5 for any additional users or groups.

Notes

- Some permissions automatically imply the selection of others. For example, the Modify Data permission for a table automatically implies the Read Data and Read Design permissions because you need these to modify the data in a table. Modify Design and Read Data imply Read Design. For [macros](#), Read Design implies Open/Run.
 - When you edit an object and save it, it retains its assigned permissions. However, if an object is saved with a new name, it is now a new object, and so has the default permissions defined for that object type rather than the permissions of the original object.
5. Exit and restart Microsoft Access and then log on as Admin.
 6. Create a new blank database, and leave it open.
 7. Import all the objects from the original database into the new database.

[How?](#)

1. Open the database or switch to the [Database window](#) for the open database.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. In the **Files Of Type** box, make sure **Microsoft Access (*.mdb; *.adp; *.mda; *.mde; *.ade)** is selected.
4. Click the arrow to the right of the **Look In** box, select the drive and folder where the [Microsoft Access database \(.mdb\)](#) or [Microsoft Access project \(.adp\)](#) that you want to [import](#) from is located, and then double-click the database's icon.
5. In the **Import Objects** dialog box, click the tab for the kind of object that you want to import, and then click each object that you want to import, or click **Select All** to import all objects of the desired type. Repeat this step for each kind of object that you want to import.

To import just the tables' definitions (not the data that they contain), click **Options**, and then under **Import Tables**, click **Definition Only**.

To import [select queries](#) as tables (for example, to create a read-only database), click **Options**, and then under **Import Queries**, click **As Tables**. [Action queries](#) import as queries no matter how this option is set.

To include [relationships](#), custom menus and toolbars, or [import/export specifications](#) (Access database only), click **Options**, and then, under **Import**, select the items you want included. Microsoft Access won't import a toolbar, menu bar, or shortcut menu if it has the same name as one in the Access file that you're importing to.

8. If users will be using the current [workgroup information file](#) when they open the database, clear the password for Admin to turn off the **Logon** dialog box for the current [workgroup](#). This is not necessary if users will be using the default workgroup information file created when they install Microsoft Access.

The new database is now completely unsecured. The workgroup information file that was current when the new database was created in step 6 defines the Admins group for the new database.

[Show All](#)

Help protect an Access database and its objects with user-level security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

[User-level security](#) is the most flexible and extensive method of implementing security measures for your [Microsoft Access database](#). You can set up security on your database to require users to log on to get into the database or not.



[Help protect a database by using the User-Level Security Wizard](#)

With the User-Level Security Wizard, you can apply [user-level security](#) with a comprehensive security scheme and encode your [Microsoft Access database](#).

1. Open the database that you want to help protect.
2. On the **Tools** menu, click **Security**, and then click **User-Level Security Wizard**.
3. Follow the directions in the wizard dialog boxes.

Notes

- The User-Level Security Wizard creates a back-up copy of the current Access database with the same name and a .bak extension, and then employs security measures for the selected objects in the current database.
- If your current Access database uses a password to help protect [Microsoft Visual Basic for Applications \(VBA\)](#) code, the wizard prompts you to unlock the Visual Basic project before the wizard will run. Unlock the project and then rerun the wizard.
- Any passwords that you create through the wizard are printed in the User-Level Security Wizard report when you finish using the wizard. You should keep this report in a secure location. You can use this report to re-create your [workgroup](#) file.

[Help protect a database without requiring users to log on](#)

If you want to help protect some parts of a database, such as the design of certain objects, but you do not care about establishing different levels of access for different groups of users, you may want to consider securing an application without requiring users to log on. This does not remove user-level security but essentially disables it for all but specific objects.

1. Help protect your database by using the User-Level Security Wizard.

[How?](#)

With the User-Level Security Wizard, you can apply [user-level security](#) with a comprehensive security scheme and encode your [Microsoft Access database](#).

1. Open the database that you want to help protect.
2. On the **Tools** menu, click **Security**, and then click **User-Level Security Wizard**.
3. Follow the directions in the wizard dialog boxes.

Notes

- The User-Level Security Wizard creates a back-up copy of the current Access database with the same name and a .bak extension, and then employs security measures for the selected objects in the current database.
 - If your current Access database protects [Microsoft Visual Basic for Applications \(VBA\)](#) code with a password, the wizard prompts you to unlock the Visual Basic project before the wizard will run. Unlock the project and then rerun the wizard.
 - Any passwords that you create through the wizard are printed in the User-Level Security Wizard report when you finish using the wizard. You should keep this report in a secure location. You can use this report to re-create your [workgroup](#) file.
2. For each table, query, form, report, and macro you want to be available to users, grant the appropriate [permissions](#) to the [Admin user account](#).

[How?](#)

1. Open the database.

The [workgroup information file](#) in use when you log on must contain the [user](#) or [group accounts](#) that you want to assign permissions for at this time; however, you can assign [permissions](#) to groups and add users to those groups later.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Permissions**.
3. On the **Permissions** tab, click **Users** or **Groups**, and then in the **User/Group Name** box, click the user or group that you want to assign permissions to.
4. Click the type of object in the **Object Type** box, and then click the name of the object to assign permissions for in the **Object Name** box. Select multiple objects in the **Object Name** box by dragging through the objects you want to select, or by holding down CTRL and clicking the objects you want.

Note Hidden objects aren't displayed in the **Object Name** box unless you select **Hidden objects** on the **View** tab of the **Options** dialog box (**Tools** menu).

5. Under **Permissions**, select the permissions you want to assign, or clear the permissions you want to remove for the group or user, and then click **Apply**. Repeat steps 4 and 5 to assign or remove permissions for additional objects for the current user or group.
6. Repeat steps 3 through 5 for any additional users or groups.

Notes

- Some permissions automatically imply the selection of others. For example, the Modify Data permission for a table automatically implies the Read Data and Read Design permissions because you need these to modify the data in a table. Modify Design and Read Data imply Read Design. For [macros](#), Read Design implies Open/Run.
- When you edit an object and save it, it retains its assigned permissions. However, if an object is saved with a new name, it is now a new object, and so has the default permissions defined for that object type rather than the permissions of the original object.

3. Turn off the **Logon** dialog box.

[How?](#)

If you don't need to establish different levels of [permissions](#) for different groups of users, you can have Microsoft Access automatically log users on as the [Admin user](#) in the [Users group](#) with the permissions specified for that group. Users will not be required to enter their user name and password in the **Logon** dialog box when they open the database. Using this method, you can help protect any or all of the objects in a database. However, each user will have the same set of permissions. This does not remove [user-level security](#) from the database.

1. Join the [workgroup](#) that has the logon procedure that you want to deactivate.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
3. Click the **Users** tab.
4. In the **Name** box, select **Admin** from the list, and then click **Clear Password**.

The next time any member of the workgroup that you joined in step 1 starts Microsoft Access and opens a database, it will no longer display the **Logon** dialog box.

[Show All](#)

Security Zones in Access

Access uses the Microsoft Internet Explorer Security Manager to determine whether a remote database is located within a trusted security zone. The Internet Explorer Security Manager is available through Internet Explorer 4.0 and later versions.



[What you need to know about security zones](#)

Internet Explorer divides your Internet world into zones, so that you can assign a Web site to a zone with a suitable security level.

Whenever you attempt to open or download a database from the Web, Microsoft Access uses the Internet Explorer Security Manager to check which security zone the database's Web site is in.

There are four different zones:

- **Internet** zone: By default, this zone contains anything that is not on your computer or an intranet, or assigned to any other zone. The default security level for the Internet zone is Medium.
- **Local intranet** zone: This zone typically contains any addresses that don't require a proxy server, as defined by the system administrator. These include sites specified on the **Connections** tab, network paths (such as \\server\share), and local intranet sites (typically addresses that don't contain periods, such as http://internal). You can assign sites to this zone. The default security level for the Local intranet zone is Medium.
- **Trusted sites** zone: This zone contains sites you trust— sites that you believe you can download or run files from without worrying about damage to your computer or data. You can assign sites to this zone. The default security level for the Trusted sites zone is Low.
- **Restricted sites** zone: This zone contains sites you don't trust— that is, sites that you're not sure whether you can download or run files from without damage to your computer or data. You can assign sites to this zone. The default security level for the Restricted sites zone is High.

In addition, any files already on your local computer are assumed to be very

safe, so minimal security settings are assigned to them. You cannot assign a folder or drive on your computer to a security zone.

Access will only open files that are located in the **Local Intranet** or **Trusted sites** zones. Access will not open files located within the **Internet** or **Restricted sites** zones. Changing the security level for a zone will not affect Access.

[Assign a Web site to a security zone](#)

1. On the **Tools** menu in Internet Explorer, click **Internet Options**.
2. Click the **Security** tab.
3. Click a security zone: **Local intranet** zone, **Trusted sites** zone, or **Restricted sites** zone.
4. Click **Sites**, and then type the Internet address for the Web site that you want to add to this zone.

Notes

- You cannot add Web sites to the **Internet** zone, which includes everything that both does not belong to any other zone and is not on your local computer.
- An Internet Protocol (IP) address for a site on an intranet must be added to the list of trusted sites in order to be recognized as trusted, even if the **Local intranet** zone is already set to the Medium or Low security level.

[Show All](#)

Create, join, or fix workgroup information files (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

A Microsoft Access [workgroup information file](#) stores information on members of a [workgroup](#), including user passwords. Access reads the workgroup information file when opening a database to determine who is allowed access to the objects in the database and what [permissions](#) they have to those objects.

Create and join a workgroup information file

 [Create a new Microsoft Access workgroup information file](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID—including whether letters are uppercase or lowercase (for all three entries)—and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

 [Join a Microsoft Access workgroup by using the Workgroup Administrator](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

[Set up more than one workgroup to use the same security-enabled database](#)

You can give users in different Microsoft Access [workgroups](#) access to a security-enabled database and its objects. This is useful if you want users at remote locations to manage their own workgroup membership.

1. If necessary, use the Workgroup Administrator to join one of the workgroups.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup**

Administrator.

3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

2. Exit Microsoft Access.
3. Restart Access, open a database, and then log on as a workgroup administrator (a member of the [Admins group](#)).
4. Create a [group account](#).

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account

and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

5. Add users to the new group.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the [workgroup](#) that contains the [user](#) and [group accounts](#).

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Open the database.

3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.

4. On the **Users** tab, enter in the **Name** box the user you want to add to a group.

5. In the **Available Groups** box, click the group you want to add the user to, and then click **Add**.

The selected group is displayed in the **Member Of** list.

6. Repeat step 5 if you want to add this user to any other groups. Repeat steps 4 and 5 to add other users to groups.

Note You can add users to this group or delete them at any time.

6. Repeat steps 1 through 5 for each workgroup that will share the same security-enabled database. In step 4, make sure to type exactly the same [case-sensitive](#) group name and [personal ID \(PID\)](#) as you did for the group account you created in the first workgroup.
7. Open the security-enabled database you want to share between the workgroups and assign [permissions](#) to the new groups.

[How?](#)

You can add or remove [permissions](#) for an existing database and its objects, or you can set what permissions are used when you create new objects.

[Assign or remove permissions for a database and its objects](#)

1. Open the database.

The [workgroup information file](#) in use when you log on must contain the [user](#) or [group accounts](#) that you want to assign permissions for at this time; however, you can assign [permissions](#) to groups and add users to those groups later.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Permissions**.
3. On the **Permissions** tab, click **Users** or **Groups**, and then in the **User/Group Name** box, click the user or group that you want to assign permissions to.
4. Click the type of object in the **Object Type** box, and then click the name of the object to assign permissions for in the **Object Name** box. Select multiple objects in the **Object Name** box by dragging through the objects you want to select, or by holding down CTRL and clicking the objects you want.

Note Hidden objects aren't displayed in the **Object Name** box unless you select **Hidden objects** on the **View** tab of the **Options** dialog box

(**Tools** menu).

5. Under **Permissions**, select the permissions you want to assign, or clear the permissions you want to remove for the group or user, and then click **Apply**. Repeat steps 4 and 5 to assign or remove permissions for additional objects for the current user or group.
6. Repeat steps 3 through 5 for any additional users or groups.

Notes

- Some permissions automatically imply the selection of others. For example, the Modify Data permission for a table automatically implies the Read Data and Read Design permissions because you need these to modify the data in a table. Modify Design and Read Data imply Read Design. For [macros](#), Read Design implies Open/Run.
- When you edit an object and save it, it retains its assigned permissions. However, if an object is saved with a new name, it is now a new object, and so has the default permissions defined for that object type rather than the permissions of the original object.

[Assign default permissions for new tables, queries, forms, reports, and macros](#)

Default permissions can be assigned only by an administrator account (a member of the [Admins group](#) in the [workgroup](#) in which the database that contains the object was created) or by the [owner](#) of the database.

1. Open the database.
2. On the **Tools** menu, point to **Security**, and then click **User And Group Permissions**.
3. On the **Permissions** tab, click **Users** or **Groups**, and then in the **User/Group Name** box, click the user or group that you want to assign permissions to.
4. Click the type of object in the **Object Type** box, and click **<New object>** in the **Object Name** list.
5. Select the default permissions that you want to assign for that object type, and then click **Apply**. Repeat steps 4 and 5 to assign default permissions for additional object types for the current user or group.

6. Repeat steps 3 through 5 for any additional users or groups, and then click **OK** when you have finished.

Note Some permissions automatically imply the selection of others. For example, the Modify Data permission for a table automatically implies the Read Data and Read Design permissions because you need these to modify the data in a table. Modify Design and Read Data imply Read Design. For [macros](#), Read Design implies Open/Run.

Note You can also have a workgroup administrator at a remote location add the same group to his or her [workgroup information file](#) by providing the administrator with the exact case-sensitive group name and PID you used to create that group in your workgroup information file.

Log on to a workgroup information file

 [Log on to a Microsoft Access workgroup](#)

The **Logon** box is displayed only if the logon procedure has been activated for your Microsoft Access [workgroup](#). Once the logon procedure has been activated, you must identify yourself at startup by entering a valid [user account](#) name. If a password has been added to a user account, you must also enter the password.

1. Start Microsoft Access.
2. If necessary, change to the workgroup that contains the user account you want to use to log on by using the Microsoft Access Workgroup Administrator.

 [How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

3. Open a database. If the logon procedure has been activated, the **Logon** box is displayed.

4. Type your user account name in the **Name** box. Your user account name isn't [case-sensitive](#).
5. If your password has been defined, type your password in the **Password** box; otherwise, leave it blank. Passwords are case-sensitive.

[Modify a workgroup information file to require users to log on to Microsoft Access](#)

When you help protect a database, you create [user accounts](#) in a Microsoft Access [workgroup](#), and then assign [permissions](#) for the database and its objects to those user accounts and to any [group accounts](#) to which they belong. Until you activate the logon procedure for a workgroup, Access automatically logs on all users at startup using the predefined [Admin user account](#). If you want users to have only the permissions associated with their own accounts, you can have them log on by using their accounts. Users log on to Microsoft Access by typing a user name and password in the **Logon** dialog box.

1. Join the workgroup whose logon procedure you want to activate.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
3. Click the **Users** tab, and make sure that the predefined Admin user account is highlighted in the **Name** box.
4. Click the **Change Logon Password** tab, and then type a new password in the **New Password** box. Don't type anything in the **Old Password** box.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

5. Verify the password by typing it again in the **Verify** box, and then click **OK**.

The **Logon** dialog box is displayed the next time any member of the workgroup that you joined in step 2 starts Microsoft Access and opens a database. Each user will use the password created in step 5 to open the database. Once in the database, they will have the permissions assigned to their group or user accounts. If no user accounts are currently defined for that workgroup, the Admin user is the only valid account at this point.

[Turn off the Logon dialog box](#)

If you don't need to establish different levels of [permissions](#) for different groups of users, you can have Microsoft Access automatically log users on as the [Admin user](#) in the [Users group](#) with the permissions specified for that group.

Users will not be required to enter their user name and password in the **Logon** dialog box when they open the database. Using this method, you can help protect any or all of the objects in a database. However, each user will have the same set of permissions. This does not remove [user-level security](#) from the database.

1. Join the [workgroup](#) that has the logon procedure that you want to deactivate.

[How?](#)

Important If you are setting up [user-level security](#) and need to make sure that your [workgroup](#) and its [permissions](#) can't be duplicated, you should make sure the [workgroup information file](#) that defines the workgroup you're joining has been created with a unique [workgroup ID \(WID\)](#). If such a workgroup information file doesn't exist, you should create one.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Join**.
4. Type the path and name of the workgroup information file that defines the Microsoft Access workgroup you want to join, and then click **OK**, or click **Browse** and then use the **Select Workgroup Information File** dialog box to locate the workgroup information file.

The next time you start Microsoft Access, it uses the [user](#) and [group accounts](#) and passwords stored in the workgroup information file for the workgroup you joined.

2. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
3. Click the **Users** tab.
4. In the **Name** box, select **Admin** from the list, and then click **Clear Password**.

The next time any member of the workgroup that you joined in step 1 starts Microsoft Access and opens a database, it will no longer display the **Logon** dialog box.

Restore or rebuild a workgroup information file

 [Restore a damaged or deleted Microsoft Access workgroup information file](#)

In rare circumstances, the Microsoft Access [workgroup information file](#) can become damaged. If you start Microsoft Access and that file can't be opened, a message will be displayed. How you solve this problem depends on whether you have a backup copy and how your workgroup information file was originally created or specified. The following table summarizes how you should address each situation.

Important Microsoft Access stores security account information in the workgroup information file. So in all cases where you don't have a backup copy and have to re-create a workgroup information file, if you've restricted [permissions](#) to your databases, you must re-create security accounts with the same [case-sensitive](#) names and [personal ID \(PID\)](#) entries as before.

Note You don't have to redefine permissions or object ownership because this information is stored in the security-enabled databases.

Used Workgroup Administrator	Made backup copy	Solution
No. Used default file created when installing Microsoft Access.	No	Re-start Microsoft Access and Microsoft Windows installer will re-create a default workgroup information file.
No. Used default file created when installing Microsoft Access.	Yes	Use Windows Explorer, My Computer, MS-DOS copy command, or backup software to copy the most recent copy of the file to the folder where you installed Microsoft Access.
Yes. Created a new file.	No	Run the Workgroup Administrator again, typing the same case-sensitive name, organization, and workgroup ID (WID) entries you used when you

Yes. Created a new file.	Yes	originally created it. Copy or restore the backup copy to the path where you originally saved your workgroup information file.
Yes. Joined (specified) a file on a path other than the folder where Microsoft Access is installed.	Yes	Copy or restore the backup copy to the original path.
Yes. Joined (specified) a file on a path other than the folder where Microsoft Access is installed.	No	Create a new copy by using the method that was used to create the original file: re-start Microsoft Access so that installer will re-create a default workgroup information file, or run the Workgroup Administrator and type the same case-sensitive name, organization, and workgroup ID entries used when it was originally created.

 [Rebuild a workgroup information file from user and group names and IDs](#)

If a [workgroup information file](#) becomes damaged or has been deleted, and a backup copy isn't available, you can re-create the workgroup information file if you have the exact [case-sensitive](#) information that you used to create the file and define the accounts and groups in the file originally.

1. Create a new workgroup information file, making sure to enter the exact case-sensitive name, company name, and [workgroup ID \(WID\)](#) that you used to create the original file. Failure to re-enter the exact entries used to create the original file will create an invalid [Admins group](#).

 [How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.

2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID— including whether letters are uppercase or lowercase (for all three entries)— and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

2. Re-create any [group accounts](#), making sure to enter the exact case-sensitive group name and [personal ID \(PID\)](#) for each group.

[How?](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

3. Re-create each [user account](#), making sure to enter the exact case-sensitive user name and PID for each user.

[How?](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
 - The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).
4. If you saved the new workgroup information file to a new name or location in step 1, tell other users in the [workgroup](#) to use the Workgroup Administrator to join the new workgroup information file.

[Show All](#)

Manage user and group accounts (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

Database [user accounts](#) provide specific privileges to individuals to access information and resources in the database. A [group account](#) contains a number of user accounts and provides a means of controlling and managing the [permissions](#) and access of this group to the objects in the database.

Create accounts, add users, and print information

[Create a security administrator account](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using a [security-enabled workgroup](#).

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID—including whether letters are uppercase or lowercase (for all three entries)—and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. To save in a different location, type a new path or click **Browse** to specify the new path.
6. Click **OK**.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

Important To help protect your database, don't use the default [workgroup](#) defined by the [workgroup information file](#) that was created when you installed Microsoft Access. You should make sure the workgroup information file that defines the workgroup you are using has been created with a unique [workgroup ID \(WID\)](#), and, if not, you should create a new file.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the administrator account and a [personal ID \(PID\)](#), and then click **OK** to create the new account.

 [Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

Caution Be sure to write down the exact account name and PID entries, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create the account, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
 - The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).
6. In the **Available Groups** box, click **Admins**, and then click **Add**.

Microsoft Access adds the new administrator account to the Admins group and displays **Admins** in the **Member Of** box.

[Create a security user account](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note It is usually easier to manage security if you organize users into groups, and then assign [permissions](#) to groups rather than to individual users.

1. Start Microsoft Access by using the [workgroup](#) in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can use the Workgroup Administrator (point to **Security** on the **Tools** menu) to find out which workgroup information file is currently in use.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.

4. On the **Users** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#), and then click **OK** to create the new account, which is automatically added to the [Users group](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you ever have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Notes

- A user account name cannot be same as an existing group account name, and visa versa.
- The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

[Create a security group account](#)

As part of securing a database, you can create [group accounts](#) in your Microsoft Access [workgroup](#) that you use to assign a common set of [permissions](#) to multiple users.

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the workgroup in which you want to use the account.

Important The accounts you create for users must be stored in the [workgroup information file](#) that those users will use. If you're using a different workgroup to create the database, change your workgroup before creating the accounts. You can use the Workgroup Administrator (point to **Security** on the **Tools** menu) to change workgroups and to find out which workgroup information file is currently in use.

2. Open a database.

3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, click **New**.
5. In the **New User/Group** dialog box, type the name of the new account and a [personal ID \(PID\)](#).

Caution Be sure to write down the exact account name and PID, including whether letters are uppercase or lowercase, and keep them in a secure place. If you have to re-create an account that has been deleted or created in a different workgroup, you must supply the same name and PID entries. If you forget or lose these entries, you can't recover them.

Note A user account name cannot be same as an existing group account name, and visa versa.

6. Click **OK** to create the new group account.

Note The PID entered in step 5 is not a password. Microsoft Access uses the PID and the user name as seeds for an encryption algorithm to generate an encrypted identifier for the [user account](#).

[Add users to security groups](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the [workgroup](#) that contains the [user](#) and [group accounts](#).

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Open the database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, enter in the **Name** box the user you want to add to a group.
5. In the **Available Groups** box, click the group you want to add the user to, and then click **Add**.

The selected group is displayed in the **Member Of** list.

6. Repeat step 5 if you want to add this user to any other groups. Repeat steps 4 and 5 to add other users to groups.

[Print information about security users and groups](#)

Note If the [workgroup information file](#) was created with Microsoft Access 95 or later, all users in the [workgroup](#) can print user and group information. If you are using a workgroup information file created with Microsoft Access version 2.0, you must be logged on as a member of the [Admins group](#) to print user and group information.

1. Start Microsoft Access by using the workgroup information file whose security information you want to print.

You can find out which workgroup information file is current or change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, click **Print Users And Groups**.
5. In the **Print Security** dialog box, do one of the following:

Click **Only Users** to print a report showing all users defined for the current workgroup.

Click **Only Groups** to print a report showing all groups defined for the current workgroup.

Click **Both Users And Groups** to print both user and group account information.

Delete accounts and remove users

[Remove users from security groups](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Notes

- You can't remove users from the default [Users group](#). Microsoft Access automatically adds all users to the Users group. To remove any user account from the Users group, you must delete the account.
- There must be at least one user in the predefined Admins group at all times.

1. Start Microsoft Access by using the [workgroup](#) containing the user and group accounts.

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Users** tab, enter the user you want to remove in the **Name** box.
5. In the **Member Of** box, click the group you want to remove the user from, and then click **Remove**.
6. Repeat step 5 to remove this user from any other groups. Repeat steps 4 and 5 to remove other users from groups.

[Delete a security user account](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note The [Admin user account](#) can't be deleted.

1. Open a database.
2. On the **Tools** menu, point to **Security**, and then click **User And Group**

Accounts.

3. On the **Users** tab, enter a user in the **Name** box, and then click **Delete**.
4. Click **Yes** to delete the user account.
5. Repeat steps 3 and 4 if you want to delete additional user accounts, and then click **OK** when you have finished.

[Delete a security group account](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

Note The Admins and [Users group](#) accounts can't be deleted.

1. Start Microsoft Access by using the [workgroup](#) that contains the account you want to delete.

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User And Group Accounts**.
4. On the **Groups** tab, enter the group you want to delete in the **Name** box, and then click **Delete**.
5. Click **Yes** to delete the group account.
6. Repeat steps 4 and 5 if you want to delete additional group accounts.

View or transfer ownership

 [View or transfer ownership of individual objects in a security-enabled database](#)

If you have Administer permission for a table, query, form, report, or macro, you can view the [owner](#) of the object or transfer its ownership to another user or group.

Note If you change ownership of a table, query, form, report, or macro to a group account, all users who belong to the group automatically receive the permissions associated with ownership of the object.

1. Open the database.
2. On the **Tools** menu, point to **Security**, and then click **User And Group Permissions**.

On the **Change Owner** tab, Microsoft Access displays a list of the tables, queries, forms, reports, and macros that are currently displayed in the [Database window](#), and the current owner of those objects.

3. Click an object type in the **Object Type** box, or use the existing object type.
4. From the **Object** list, click one or more objects whose ownership that you want to change. To select more than one object, either hold down CTRL and click the objects, or drag through the ones you want to select.
5. In the **New Owner** box, click the user or group account that you want to be the new owner of the object or objects.
6. Click the **Change Owner** button.

 [Transfer ownership of an entire database to another administrator](#)

Note To import a database, you must have Open/Run permission for the database, and Read Design permission for its objects. To import tables, you must also have Read Data permission. If you have permissions for some tables, queries, forms, reports, and macros but not others, Microsoft Access imports only those objects for which you have permissions.

1. Start Microsoft Access by using a [security-enabled workgroup](#) that contains the user account that you want to own the database and its objects.

[How?](#)

A Microsoft Access [workgroup information file](#) contains a list of users who share data, also known as a [workgroup](#). Users' passwords are also stored in the workgroup information file. To control who has access to your database, you must create a new workgroup information file.

1. Start Microsoft Access.
2. On the **Tools** menu, point to **Security**, and then click **Workgroup Administrator**.
3. In the **Workgroup Administrator** dialog box, click **Create**.
4. In the **Workgroup Owner Information** dialog box, type your name and organization, and then type any combination of up to 20 numbers and letters for the [workgroup ID \(WID\)](#).

Caution Be sure to write down your exact name, organization, and workgroup ID—including whether letters are uppercase or lowercase (for all three entries)—and keep them in a secure place. If you have to re-create the workgroup information file, you must supply exactly the same name, organization, and workgroup ID. If you forget or lose these entries, you can't recover them and might lose access to your databases.

5. Type a new name for the new workgroup information file. By default, the workgroup information file is saved in the language folder. To save in a different location, type a new path or click **Browse** to specify the new path.

The new workgroup information file is used the next time you start Microsoft Access. Any [user](#) and [group accounts](#) or passwords that you create are saved in the new workgroup information file. To have others join the workgroup defined by your new workgroup information file, copy the file to a shared folder (if you didn't already save it in a shared folder in step 5), and then have each user run the Workgroup Administrator to join the new workgroup information file.

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Log on by using that account.
3. Create a new blank database.
4. Import all of the objects into the new database from the database that has the ownership you want to change.

[How?](#)

1. Open the database or switch to the [Database window](#) for the open database.
2. On the **File** menu, point to **Get External Data**, and then click **Import**.
3. In the **Files Of Type** box, make sure **Microsoft Access (*.mdb; *.adp; *.mda; *.mde; *.ade)** is selected.
4. Click the arrow to the right of the **Look In** box, select the drive and folder where the [Microsoft Access database \(.mdb\)](#) or [Microsoft Access project \(.adp\)](#) that you want to [import](#) from is located, and then double-click the database's icon.
5. In the **Import Objects** dialog box, click the tab for the kind of object that you want to import, and then click each object that you want to import, or click **Select All** to import all objects of the desired type. Repeat this step for each kind of object that you want to import.

To import just the tables' definitions (not the data that they contain), click **Options**, and then under **Import Tables**, click **Definition Only**.

To import [select queries](#) as tables (for example, to create a read-only database), click **Options**, and then under **Import Queries**, click **As Tables**. [Action queries](#) import as queries no matter how this option is set.

To include [relationships](#), custom menus and toolbars, or [import/export specifications](#) (Access database only), click **Options**, and then, under **Import**, select the items you want included. Microsoft Access won't import a toolbar, menu bar, or shortcut menu if it has the same name as one in the Access file that you're importing to.

[Show All](#)

Types of permissions (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

The following table summarizes the [permissions](#) that you can assign.

Permission	Permits a user to
Open/Run	Open a database, form, or report, or run a macro in a database.
Open Exclusive	Open a database with exclusive access.
Read Design	View tables, queries, forms, reports, or macros in Design view .
Modify Design	View and change the design of tables, queries, forms, reports, or macros; or delete them. For databases, set a database password, replicate a database, and change startup properties.
Administer	For tables, queries, forms, reports, and macros, have full access to these objects and data, including ability to assign permissions.
Read Data	View data in tables and queries.
Update Data	View and modify, but not insert or delete, data in tables and queries.
Insert Data	View and insert, but not modify or delete, data in tables and queries.
Delete Data	View and delete, but not modify or insert, data in tables and queries.

Notes

- Some permissions automatically imply the selection of others. For example, the Update Data permission for a table automatically implies the Read Data and Read Design permissions because you need these to modify the data in a table. Modify Design and Read Data imply Read Design. For macros,

Read Design implies Open/Run.

- To design forms, reports, macros, and [modules](#) in a multiuser environment, you must open a [Microsoft Access database](#) exclusively, which means you must have Open Exclusive permission on the Access database.
- In general, if you want users to be able to access a linked table, grant them Read Data and Read Design permissions on the table in the back-end database, and Modify Design permission on the table link defined in the front-end database so that they can conveniently re-link the table. If you want to restrict all access to the back-end table but still allow users to view the data and re-link the tables, remove all permissions on the back-end table and use queries in the front-end database with their **RunPermissions** property set to **Owner's**.

[Show All](#)

About securing a replicated database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To help protect a [replicated](#) database, you can set user [permissions](#) on replicated database objects. These permissions will not interfere with [synchronization](#) of the database. However, you can't use a password to protect your replicated database. A password-protected database can't be made replicable, nor can a password be assigned to a replicated database.

You can also keep a [replica](#) in an unshared directory and use [indirect synchronization](#). If [direct synchronization](#) is required, set the permissions on the server where the replicas reside so that only specific users can access them. Do not put replicas on read-only servers, because synchronizations with other [replica set](#) members will not occur.

[Show All](#)

About securing Visual Basic for Applications code in Access databases (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can help protect your [Microsoft Visual Basic for Applications \(VBA\)](#) code by using either a password or by saving your database as an MDE file. A password will only help protect the code, while saving your database as an MDE file will help protect your forms, reports, and Visual Basic code. [User-level security](#) will not secure your code, but it will help protect the forms and reports that refer to the code.



[Securing Visual Basic for Applications \(VBA\) code with a password](#)

In a [Microsoft Access project \(.adp\)](#) file and a [Microsoft Access database \(.mdb\)](#) file, you can help protect all [standard modules](#) and [class modules](#) (such as code behind forms and reports) by employing passwords for VBA code, which you and other users must enter to view or edit VBA code in the Visual Basic Editor. Once you set a password, you enter this password once per session. The password is required not only for viewing and editing, but also for cutting, copying, pasting, exporting, and deleting any module. Note, however, that employing security measures for your VBA code in this manner doesn't prevent you or other users from running existing VBA code.



[Saving your database as an MDE file](#)

Saving your [Microsoft Access database](#) as an MDE file compiles all modules, removes all editable source code, and compacts the destination database. Your Visual Basic code will continue to run, but it cannot be viewed or edited. Your Access database will continue to function normally— -you can still update data and run reports— -and the size of the database will be reduced due to the removal of the code. Memory usage is optimized, which will improve

performance.

Saving your Access database as an MDE file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in Design view.
- Adding, deleting, or changing references to object libraries or databases.
- Changing code using the properties or methods of the Microsoft Access or VBA Object models— an MDE file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, queries, [data access pages](#), and macros can be imported from or exported to non-MDE databases.

[Modules and user-level security](#)

In a Microsoft Access 2000 or later database, [standard modules](#) and [class modules](#) (such as code behind forms and reports) no longer employ user-level security, as they did in previous versions. However, forms and reports still make use of user-level security. This means that modules and forms and reports employ different security mechanisms.

For example, assuming you have the appropriate [permissions](#), you can add a control to a form, but if the VBA project is locked for viewing and protected with a password, and you don't know the password, you can't view or edit the module behind the form. Conversely, you can be restricted from designing a form or report through user-level security by not having Modify Design permission on the object, but if you know the VBA project password, you can access the module behind the form or report.

There is an exception to this. Even if you have Modify Design permission on a form or report, you cannot delete the form or report, or set the **HasModule** property to **No**, because this action deletes the module behind the form or report.

[Show All](#)

Troubleshoot Visual Basic code security (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

 [I get the message "Microsoft Access was unable to create an MDE database" when I try to save my database as an MDE file.](#)

A replicated database (either a [replica](#) or [Design Master](#)) cannot be saved as an MDE file. To save a replicated database as an MDE file, you must first remove [replication](#).

Once a database is saved as an MDE file, it can be replicated; however, replication is only recommended in situations where no further changes need to be made to the original database. There is no way to make changes to the design of forms, reports, or code in the Design Master MDE file, and design changes made in the original database can't be [synchronized](#) with pre-existing MDE replicas. For this reason, if you need to make a design change to a [replica set](#) of MDE files, you must make that change in the original database, resave it as an MDE file, and then create and distribute an entirely new replica set from the new MDE file.

I forgot the password to my Microsoft Visual Basic code.

If you forget the password, you will not be able to view or edit the Microsoft [Visual Basic for Applications \(VBA\)](#) code. There is no way to recover the password from the database.

[Show All](#)

Remove a Visual Basic for Applications code password

1. Open the [Microsoft Access project \(.adp\)](#) file or [Microsoft Access database \(.mdb\)](#) file.
2. In the [Database window](#), point to **Macro** on the **Tools** menu, and then click **Visual Basic Editor**.



You can also press ALT+F11.

3. In the Microsoft Visual Basic Editor, click <**Access *database or Access project name***> **Properties** on the **Tools** menu.
4. Click the **Protection** tab, and then clear all information.

Your code is now unsecured. You can add a password whenever you need to help protect your code.

About MDE and ADE files

[About MDE files \(MDB\)](#)

[About ADE files \(ADP\)](#)

[Show All](#)

About MDE files (MDB)

If your database contains [Microsoft Visual Basic for Applications \(VBA\)](#) code, saving your [Microsoft Access database](#) as an MDE file compiles all modules, removes all editable source code, and compacts the destination database. Your Visual Basic code will continue to run, but it cannot be viewed or edited. Saving your database as an MDE file helps protect your forms and reports without requiring users to log on or requiring you to create and manage the [user accounts](#) and [permissions](#) that are needed for [user-level security](#). Your database will continue to function as it did— you can still update data and run reports.

Saving your Access database as an MDE file prevents the following actions:

- Viewing, modifying, or creating forms, reports, or modules in Design view.
- Adding, deleting, or changing references to object libraries or databases.
- Changing code— an MDE file contains no source code.
- Importing or exporting forms, reports, or modules. However, tables, queries, [data access pages](#), and macros can be imported from or exported to non-MDE databases.

[Considerations if you need to modify the design of forms, reports, or modules](#)

Be sure to keep a copy of your original Access database. If you need to modify the design of forms, reports, or modules in an MDE file, you must modify the original Access database, and then resave it as an MDE file. Saving an Access database containing tables as an MDE file creates complications reconciling different versions of the data if you need to modify the design of the forms, reports, or modules later. For this reason, saving an Access database as an MDE file is most appropriate for the front-end database of a [front-end/back-end application](#).

[Code no longer employs user-level security](#)

In previous versions of Access, user-level security could be used with VBA modules. In Microsoft Access 2000 and later, all Visual Basic code for an

Access database (.mdb) file or an [Access project \(.adp\)](#) file, including [stand-alone modules](#) and [class modules](#) (such as code behind forms and reports) must employ security technologies such as setting a password, or by saving the database as an MDE or ADE file, which removes the source code.

[Consideration before saving your database as an MDE file](#)

Some restrictions may prevent you from saving your Access database as an MDE file:

- You must have password access to the Visual Basic code.
- If your database is replicated, you must first remove [replication](#).
- If your Access database references another Access database, or [add-in](#), you must save all Access databases or add-ins in the chain of references as MDE files.

Additionally, if you define a database password or user-level security before saving an Access database as an MDE file, those features will still apply to an MDE file created from that database. If your Access database has a database password or user-level security defined and you want to remove these features, you must do so before saving it as an MDE file.

To save an Access database that employed user-level security as an MDE file, you must meet the following requirements before you can proceed:

- You must join the [workgroup information file](#) that defines the user accounts used to access the database, or that were in use when the database was created.
- Your user account must have Open/Run and Open Exclusive permissions for the database.
- Your user account must have Modify Design or Administer permissions for any tables in the database, or you must be the [owner](#) of any tables in the database.
- Your user account must have Read Design permissions for all objects in the database.

[About references and MDE files](#)

If you try to create an MDE file from a Microsoft Access database (.mdb) or an

[add-in \(.mda\)](#) that references another Access database or add-in, Access displays an error message and doesn't let you complete the operation. To save a database that references another database as an MDE file, you must save all databases in the chain of references as MDE files, starting from the first database referenced. After saving the first database as an MDE file, you must then update the reference in the next database to point to the new MDE file before saving it as an MDE file, and so on.

For example, if Database1.mdb references Database2.mdb, which references Database3.mda, you would proceed as follows:

1. Save Database3.mda as Database3.mde.
2. Open Database 2.mdb and change its reference to point to the new Database3.mde.
3. Save Database2.mdb as Database2.mde.
4. Open Database1.mdb and change its reference to point to the new Database2.mde.
5. Save Database1.mdb as Database1.mde.



[About saving a replicated database as an MDE file](#)

A replicated database (either a [replica](#) or [Design Master](#)) cannot be saved as an MDE file. To save a replicated database as an MDE file, you must first remove [replication](#).

Once a database is saved as an MDE file, it can be replicated; however, replication is only recommended in situations where no further changes need to be made to the original database. There is no way to make changes to the design of forms, reports, or code in the Design Master MDE file, and design changes made in the original database can't be [synchronized](#) with pre-existing MDE replicas. For this reason, if you need to make a design change to a [replica set](#) of MDE files, you must make that change in the original database, resave it as an MDE file, and then create and distribute an entirely new replica set from the new MDE file.

[Show All](#)

About passwords (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

You can use three types of passwords in Microsoft Access. The type of password protection you choose will determine the level of user access to your database and the objects it contains.

[Database passwords](#)

If you use a database password, all users must enter that password before they are allowed to open the database. Adding a database password is an easy way to help prevent unwanted users from opening your database; however, once a database is open, no other security measures are provided unless [user-level security](#) has been defined as well.

Microsoft Access stores the database password in an unencrypted form. If this will compromise the security of the password-protected database, you should not use a database password to protect the database. Instead, you should define user-level security to help control access to sensitive data in that database.

[Security account passwords](#)

When user-level security has been defined for a [workgroup](#), you can use a security account password. A security account password helps prevent unauthorized users from logging on using another user's name.

By default, Microsoft Access assigns a blank password to the default [Admin user account](#) and to any new [user accounts](#) that you create in your workgroup. As part of securing a database, it's important to add a password to:

- The Admin user account (to activate the **Logon** dialog box).
- The user account that owns the database and its tables, queries, forms, reports, and macros.
- Any user accounts that you add to the [Admins group](#).

In addition, you might want to add passwords to the accounts you create for users, or instruct users to add their own passwords.

Users can create or change their own user account passwords; however, only an administrator account can clear a password if a user forgets the password.

[Microsoft Visual Basic for Applications \(VBA\) passwords](#)

In addition, you can also set a [Visual Basic for Applications \(VBA\)](#) password. You use this password to protect VBA code in [standard modules](#) and [class modules](#) (such as code behind forms and reports). This password is entered when you first attempt to open any VBA code and prevents unauthorized users from editing, cutting, pasting, copying, exporting, and deleting the code.

[Show All](#)

Create, change, or remove passwords (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).



[Set a password in an Access database \(.mdb\)](#)

You can provide limited protection for your database by adding a password to restrict which users can open the database. Microsoft Access stores the database password in an unencrypted form. If this will compromise the security of sensitive data in the database, you should use [user-level security](#) instead of a database password.

Caution

- If you lose or forget your password, it can't be recovered, and you won't be able to open your database.
 - Do not use a database password if you will be [replicating](#) a database. Replicated databases can't be [synchronized](#) if database passwords are defined.
 - You can't set a database password if user-level security has been defined for your database and you don't have Administer [permission](#) for the database. Also, a database password is defined in addition to user-level security. If user-level security has been defined, any restrictions based on user-level security permissions remain in effect.
1. Close the database. If this is a shared database, make sure all other users have closed the database.
 2. Make a backup copy of the database, and store it in a secure place.
 3. On the **File** menu, click **Open**.
 4. Click the arrow to the right of the **Look In** box, select the drive and folder where the [Microsoft Access database \(.mdb\)](#) is located, and then click the database's icon.
 5. Click the arrow to the right of the **Open** button, and then click **Open**

Exclusive.

6. On the **Tools** menu, point to **Security**, and then click **Set Database Password**.
7. In the **Password** box, type your password.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

8. In the **Verify** box, confirm your password by typing the password again, and then click **OK**.

The password is now set. The next time you or any other user opens the database, a dialog box will be displayed that requests a password.

Notes

- A database password is stored with the database and not with the [workgroup information file](#).
- If a table from a password-protected database is linked, the password is cached (saved) in the database it is linked to when the link is established. This will allow anyone to see your data.

[Set a password in an Access project \(.adp\)](#)

Unlike a [Microsoft Access database](#), you can't help protect forms, reports, or

macros in an Access project by using [user-level security](#), nor can you set a password on the [Microsoft Access project \(.adp\)](#) file. To help protect form and report objects, you can hide the objects in the Database window or set startup options. To help protect access to the design of forms and reports in an Access project file, you can either set startup options or save your Access project file as an .ade file. To help protect access to macros in an Access project file, use startup options. You can help protect a data access page by using file or folder security for your operating system. You can also help protect your Visual Basic for Applications code by converting your Access project file to an .ade file or by setting a password.

[Show or hide database objects in the Database window](#)

[Specify whether a database object is hidden or visible](#)

1. Under **Objects** in the [Database window](#), click the type of database object whose properties you want to change.
2. Click **Properties**  on the **Database** toolbar.
3. Select or clear the **Hidden** check box.

Note In a [Microsoft Access project](#), you cannot change the properties of a table, query, or [database diagram](#) because these objects reside in the Microsoft SQL Server database. However, you can change the properties of a form, report, macro, or module, because these objects reside in the Access project itself, not in the Microsoft SQL Server database. You can also change the properties of a [data access page](#).

[Show or hide objects defined as hidden by default](#)

If you need to work with objects that are defined as hidden without changing their status as hidden, you can display them in the [Database window](#).

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **Hidden objects** check box. Icons for hidden objects appear dimmed to distinguish them from objects that aren't defined as hidden.

[Show or hide system objects by default](#)

Microsoft Access automatically creates [system objects](#) when you create a new database.

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show**, select or clear the **System objects** check box.

[Use startup options](#)

You can use startup options to specify, for example, what form to display, whether toolbars can be customized, and whether [shortcut menus](#) are available in your [Microsoft Access file](#). You can also use a special [macro](#) named AutoExec to carry out an [action](#) or series of actions when your database first opens. When you open a database, Microsoft Access looks for a macro with this name and, if it finds one, runs it automatically.

Set startup options

1. On the **Tools** menu, click **Startup**.
2. Select the options or enter the settings you want to use.

[Create a macro that runs when an Access file first opens](#)

1. Create a [macro](#) containing the actions you want to run when you open the database.
2. Save the macro with the name AutoExec.

The next time you open the database, Microsoft Access runs this macro automatically.

Note If you don't want to run the AutoExec macro when you open the database, hold down the SHIFT key when the database opens.

[Help protect data access pages](#)

Data access pages are [Hypertext Markup Language \(HTML\)](#) files that reference the data in a database. The data access pages aren't actually stored in the [Microsoft Access file](#). For this reason, Access has no control over the security of the data access page files. To help protect a data access page that is stored on a local or network file system, assuming you have the appropriate [permissions](#), you can use the following procedure.

Note To help protect data access page files that are stored on an [HTTP](#) server, you must use the security features available on the server itself.

1. Open **Windows Explorer**  or **My Computer**.
2. Locate the folder where the data access page's HTML file is located. By default, the file will be located in the same directory as your Access database.
3. Right-click the data access page (.htm) file or the folder containing the file, click **Properties** on the shortcut menu, and then select the **Read-only** check box.

[Help protect Visual Basic for Applications code](#)

You can help protect the VBA code in your [Microsoft Access project \(.adp\)](#) using a password or by saving a copy of your file as an [.ade](#) file. Saving your .adp file as an .ade file provides greater security than password protecting your code. However, there are some limitations in using an .ade file.

[Help protect Visual Basic for Applications code with a password](#)

To prevent someone from viewing or making changes to your [Microsoft Visual Basic for Applications \(VBA\)](#) code, you can help protect the code by requiring a password.

1. Open the [Microsoft Access project \(.adp\)](#) file or [Microsoft Access database \(.mdb\)](#) file that has the VBA code you want to help protect.
2. In the Database window, point to **Macro** on the **Tools** menu, and then click **Visual Basic Editor**.

[Tip](#)

You can also press ALT+F11 to open the **Visual Basic Editor**.

3. In the Visual Basic Editor, click **<Access database or Access project name> Project Properties** on the **Tools** menu.
4. On the **Protection** tab, select the **Lock project for viewing** check box. If you set a password but don't select **Lock project for viewing**, the code can be viewed and edited by anyone, but the **Project Properties** dialog box continues to employ its security features.
5. In the **Password** box, type your password.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the

following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

6. In the **Confirm password** box, confirm your password by typing the password again, and then click **OK**.

The password is now set. The next time you or any other user opens the database, a dialog box will be displayed that requests a password.

Important If you forget your password, it can't be recovered, and you won't be able to view or edit the VBA code.

 [Make an ADE file](#)

Caution

Be sure to save a copy of your original [Microsoft Access project \(.adp\)](#) file.

- You can't modify the design of forms, reports, or modules in a Access project file saved as an .ade file. If you need to change the design of these objects, you must do so in the original Access project file and then resave the Access project file as an .ade file.
 - You can't create an .ade file in Access 2002 or later from an Access project file created in an earlier version. You can only save an .ade file from an Access project file from the same version of Access.
 - You won't be able to convert an Access project file saved as an .ade file in future versions of Microsoft Access. You will be able to run the .ade file in a later version of Access.
1. Close the Access project file. If you are working in a [multiuser](#) environment, make sure that all other users have closed the Access project file.
 2. On the **Tools** menu, click **Database Utilities**, and then click **Make ADE File**.
 3. In the **Database To Save As ADE** dialog box, specify the Access project

file you want to save as an .ade file, and then click **Make ADE**.

4. In the **Save ADE As** dialog box, specify a name, drive, and folder for the Access project.

[Remove a database password in an Access database](#)

1. On the **File** menu, click **Open**.
2. Click the arrow to the right of the **Open** button, click **Open Exclusive**, and then open the database.
3. In the **Password Required** dialog box, type the database password, and then click **OK**. Passwords are [case-sensitive](#).
4. On the **Tools** menu, point to **Security**, and then click **Unset Database Password**. This command is only available if a database password was set previously.
5. In the **Unset Database Password** dialog box, type your current password.

[Create or change a security account password in an Access database](#)

A security account password is created to make sure that no user can log on using another user name. By default, Microsoft Access assigns a blank password to the [Admin user account](#), and to any new [user accounts](#) you create in your [workgroup](#).

Caution You can't recover your password if you forget it, so be sure to store it in a safe place. If you forget your password, a user logged on with an administrator account (a member of the [Admins group](#) in the workgroup in which the account and password were created) must clear the password before you can log on.

1. Start Microsoft Access by using the workgroup the user account is stored in, and log on using the name of the account for which you want to create or change the password.

You can find out which workgroup is current or change workgroups by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User and Group Accounts**.

4. On the **Change Logon Password** tab, leave the **Old Password** box blank if a password hasn't been defined previously for this account. Otherwise, type the current password in the **Old Password** box.
5. Type the new password in the **New Password** box.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

6. Retype the password in the **Verify** box, and then click **OK**.

[Change a password in an Access project](#)

If your [Microsoft Access project \(.adp\)](#) is connected to a Microsoft SQL Server 6.5 (or later) database with SQL Server security implemented, then you can change the logon password from within Access.

1. Open an Access project file or switch to the [Database window](#) for the open project.
2. On the **Tools** menu, point to **Security**, and then click **Set Login Password**.
3. In the **Old Password** box, type the current password.
4. In the **New Password** box, type your password.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers,

and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

5. In the **Verify** box, confirm your password by typing the password again, and then click **OK**.

The password is now changed and Access resets the .adp connection with the new password information. The next time you or any other user opens the project, a dialog box will be displayed that requests the new password.

Important If you lose or forget your password, it can't be recovered, and you won't be able to open your project.

[Clear a security account password](#)

To complete this procedure, you must be logged on as a member of the [Admins group](#).

1. Start Microsoft Access by using the [workgroup information file](#) in which the [user account](#) is stored.

You can find out which workgroup information file is current or change [workgroups](#) by using the Workgroup Administrator.

2. Open a database.
3. On the **Tools** menu, point to **Security**, and then click **User and Group Accounts**.
4. On the **Users** tab, enter the user account name in the **Name** box.
5. Click **Clear Password**.

6. Repeat steps 4 and 5 to clear any additional passwords, and then click **OK** when you have finished.

[Help protect Microsoft Visual Basic for Applications \(VBA\) code with a password](#)

To prevent someone from viewing or making changes to your [Microsoft Visual Basic for Applications \(VBA\)](#) code, you can help protect the code by requiring a password.

1. Open the [Microsoft Access project \(.adp\)](#) file or [Microsoft Access database \(.mdb\)](#) file that has the VBA code you want to help protect.
2. In the Database window, point to **Macro** on the **Tools** menu, and then click **Visual Basic Editor**.

[Tip](#)

You can also press ALT+F11 to open the **Visual Basic Editor**.

3. In the Microsoft Visual Basic Editor, click **<Access database or Access project name> Project Properties** on the **Tools** menu.
4. On the **Protection** tab, select the **Lock project for viewing** check box. If you set a password but don't select **Lock project for viewing**, the code can be viewed and edited by anyone, but the **Project Properties** dialog box continues to employ its security features.
5. In the **Password** box, type your password.

[Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *

- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

6. In the **Confirm password** box, confirm your password by typing the password again, and then click **OK**.

The password is now set. The next time you or any other user opens the database, a dialog box will be displayed that requests a password.

Important If you forget your password, it can't be recovered, and you won't be able to view or edit the VBA code.

[Show All](#)

About linking tables from a password-protected database (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

To link a table from a [Microsoft Access database](#) that is password-protected, you must supply the correct [password](#). If you supply the correct password, Microsoft Access stores the database password with the information that defines the link to the table. After the link has been defined, any user who can open the database that the table is linked to can open the linked table. When a user opens the database with the link to the table, Microsoft Access uses the stored password to open the database where the table is stored. If the password is changed for the database where the table is stored, the next time the database with the link to the table is opened, the new password must be supplied before Microsoft Access will open it.

Microsoft Access stores the database password information for the linked table in an unencrypted form. If this will compromise the security of the password-protected database, you should not use a database password to help protect the database. Instead, you should define [user-level security](#) to help control access to sensitive data in that database.

[Show All](#)

Troubleshoot passwords (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

I can't delete my security account password.

Only an administrator account (a member of the [Admins group](#) used to manage security for your [workgroup](#)) can delete a security account password. If you have forgotten or lost your password, you must get an administrator to delete it.

I forgot my password.

You can't recover your password. If you forget your password, a user logged on with an administrator account (a member of the Admins group in the workgroup in which the account and password were created) must clear the password before you can log on.

[Show All](#)

About digital signatures

Microsoft Office 2003 uses Microsoft Authenticode technology to enable you to [digitally sign](#) a file or [macro project](#) by using a [digital certificate](#). The certificate used to create this signature confirms that the macro originated from the signer, and the signature confirms that it has not been altered. When you set the macro security level, you can run macros based on whether they are digitally signed by a developer on your list of trusted sources.



[Where to get the digital certificates](#)

You can obtain a digital certificate from a commercial certification authority, such as VeriSign, Inc., or from your internal security administrator or Information Technology (IT) professional. Or, you can create a digital signature yourself using the Selfcert.exe tool.

Note Because a digital certificate you create yourself isn't issued by a formal certification authority, macro projects signed by using such a certificate are referred to as self-signed projects. Certificates you create yourself are considered unauthenticated and will generate a warning in the **Security Warning** box if the security level is set to High or Medium. Microsoft Office will only trust a self-signed certificate on a computer that has the private key for that certificate available (generally only the computer that actually created the certificate, unless the private key is shared with other computers).

Commercial certification authorities

To obtain a digital certificate from a commercial certification authority, such as VeriSign, Inc., you or your organization must submit an application to that authority.

To learn more about certification authorities that offer services for Microsoft products, see the Microsoft TechNet Security Web site.

Depending on your status as a developer, you should apply for a Class 2 or Class 3 digital certificate for software publishers:

- A Class 2 digital certificate is designed for people who publish software as individuals. This class of digital certificate provides assurance as to the identity of the individual publisher.
- A Class 3 digital certificate is designed for companies and other organizations that publish software. This class of digital certificate provides greater assurance about the identity of the publishing organization. Class 3 digital certificates are designed to represent the level of assurance provided today by retail channels for software. An applicant for a Class 3 digital certificate must also meet a minimum financial stability level based on ratings from Dun & Bradstreet Financial Services.

When you receive your digital certificate, you are given instructions on how to install it on the computer you use to sign your Microsoft Office solutions.

Internal certification authorities

Some organizations and corporations might have a security administrator or group act as their own certification authority and produce or distribute digital certificates by using tools such as Microsoft Certificate Server. Microsoft Certificate Server can function as a stand-alone certification authority or as part of an existing certification authority hierarchy. Depending on how Microsoft Office digital-signature features are used in your organization, you might be able to sign macro projects by using a digital certificate from your organization's internal certification authority. Or you might need to have an administrator sign macro projects for you by using an approved certificate. For information about your organization's policy, contact your network administrator or IT department.



[Signing your own files and macro projects](#)

After you have installed your digital certificate, you can sign macro projects.

When you digitally sign a file, you certify that the information in the file is valid and that it has not been modified since the file was signed. As long as a file is unchanged, reviewers can attach signatures to it. You might use a digital signature with important files. When you digitally sign a macro project, your digital signature says that you guarantee that the project is safe. Just as signed files remain signed until the file is modified, signed macro projects remain signed until the macro code is altered.

[Show All](#)

Add a digital signature to macro project

You [digitally sign](#) a file or a [macro project](#) by using a [digital certificate](#).

1. If you don't already have a digital certificate, you must obtain one.



You can obtain a digital certificate from a commercial certification authority, such as VeriSign, Inc., or from your internal security administrator or Information Technology (IT) professional. Or, you can create a digital signature yourself using the Selfcert.exe tool.

To learn more about certification authorities that offer services for Microsoft products, see the list of [Microsoft Root Certificate Program Members](#).

Notes

- The [hyperlink](#) in this topic goes to the Web. You can switch back to Help at any time.
 - Because a digital certificate you create yourself isn't issued by a formal certification authority, macro projects [signed](#) by using such a certificate are referred to as self-signed projects. Depending on how Microsoft Office digital-signature features are being used in your organization, you might be prevented from using such a certificate, and other users might not be able to run self-signed macros for security reasons.
2. Open the file that contains the macro project you want to sign.
 3. On the **Tools** menu, point to **Macro**, and then click **Visual Basic Editor**.
 4. In the **Project Explorer**, select the project you want to sign.
 5. On the **Tools** menu, click **Digital Signature**.
 6. If you haven't previously selected a digital certificate or want to use another

- one, click **Choose**, select the certificate, and then click **OK** twice.
7. To use the current certificate, click **OK**.

Tips

- Sign macros only after your solution has been tested and is ready for distribution, because whenever code in a signed macro project is modified in any way, its digital signature is removed. However, if you have the proper digital certificate on your computer, the macro project will automatically be resigned when saved.
- If you want to prevent users of your solution from accidentally modifying your macro project and invalidating your signature, lock the macro project before signing it. Your digital signature says only that you guarantee that the project has not been tampered with since you signed it. It does not prove that you wrote the project. Locking your macro project doesn't prevent another user from replacing the digital signature with another signature. Corporate administrators might resign [templates](#) and [add-ins](#) so that they can control exactly what users may run on their computers.
- If you create an add-in that adds code to a macro project, your code should determine if the project is digitally signed and notify the user of the consequences of modifying a signed project before continuing.
- When digitally signing macros, consider obtaining a timestamp so that others can verify your signature even after the certificate used for the signature has expired. See Microsoft Office Online for more information on macro security and timestamps.

[Show All](#)

Remove a digital signature

1. Open the file that contains the [macro project](#) from which you want to remove the signature.
2. On the **Tools** menu, point to **Macro**, and then click **Visual Basic Editor**.
3. In the **Project Explorer**, select the project from which you want to remove the signature.
4. On the **Tools** menu, click **Digital Signature**.
5. Click **Remove**.

[Show All](#)

Create your own digital certificate

Because a [digital certificate](#) you create yourself isn't issued by a formal certification authority, [macro projects signed](#) by using such a certificate are referred to as self-signed projects. Certificates you create yourself are considered unauthenticated and will generate a warning in the **Security Warning** box if the security level is set to High or Medium. Microsoft Office will only trust a self-signed certificate on a computer that has the private key for that certificate available (generally only the computer that actually created the certificate, unless the private key is shared with other computers).

To learn more about certification authorities that offer services for Microsoft products, see the list of [Microsoft Root Certificate Program Members](#).

Note The [hyperlink](#) in this topic goes to the Web. You can switch back to Help at any time.

1. In Microsoft Windows XP, on the Windows **Start** menu, point to **All Programs**, point to **Microsoft Office**, point to **Microsoft Office Tools**, and then click **Digital Certificate for VBA Projects**. In Microsoft Windows 2000, on the Windows **Start** menu, point to **Programs**, point to **Microsoft Office Tools**, and then click **Digital Certificate for VBA Projects**.

If the SelfCert.exe file is not on your computer, you might need to install it.

[How?](#)

1. Run the Microsoft Office Setup program again.

[How?](#)

1. Quit all programs.
2. Double-click the **Add/Remove Programs** icon in the Microsoft Windows **Control Panel**.
3. Do one of the following:
 - If you installed your Office program as part of Microsoft Office, click **Microsoft Office** in the **Currently installed**

- programs** box, and then click the **Change** button.
- If you installed your Office program individually, click the name of your program in the **Currently installed programs** box, and then click the **Change** button.
2. On the **Advanced Customization** screen in the setup program, click the expand indicator (+) next to **Office Shared Features**.
 3. Select **Digital Signature for VBA projects**, click the arrow next to your selection, and then click **Run from My Computer**.
2. Follow the instructions in the **Create Digital Certificate** dialog box.

[Show All](#)

Review digital certificates for a macro project

When you look at the details of a [digital certificate](#), look at the **Issued to** and **Issued by** fields to determine whether you trust its source. Look at the **Valid from** field to determine if the certificate is current.

If your **Macro Security** setting is set to Medium or High, you will receive a warning before you open files that contain [macros signed](#) with suspicious certificates. The warning dialog box allows you to click **Details** to view the properties of the certificate.

To view certificates for a file that you already have open, follow these steps:

1. On the **Tools** menu, point to **Macro**, and then click **Visual Basic Editor**.
2. Use the project explorer to select the desired macro project.
3. In the **Visual Basic Editor**, click **Digital Signature** on the **Tools** menu.
4. In the **Digital Signature** dialog box, click **Detail** to view the details of the certificate.

Note The **Detail** button will only appear if a digital certificate is attached to the project.

[Show All](#)

About helping protect files from macro viruses

A macro virus is a type of computer [virus](#) that's stored in a [macro](#), Visual Basic for Applications (VBA) code, a [form](#) that contains an [ActiveX control](#), a [report](#) that contains an ActiveX control, [action query](#), [data-definition query](#), [pass-through query](#), or [update query](#). To help protect against macro viruses, you should purchase and install specialized antivirus software.

For more information about using antivirus software with Microsoft Office 2003, see the Microsoft Office Online Web site.

To further help reduce the risk of macro infection in Office files, set the macro security level to **High** or **Medium** and use [digital signatures](#).

 [Security levels for macros](#)

The levels of security to help reduce macro virus infection are as follow:

- **Very High** This setting is not available in the **Security** dialog box in Microsoft Office Access 2003. It is possible, however, to use system policies to set the security level in Access to **Very High**. When the security level is set to **Very High**, Access cannot open any [Access database](#) or [Access project](#) files.
- **High** You can open files that have been digitally signed and that you confirm are from a trusted source Before deciding to trust a source, you should confirm that the source is responsible and uses a virus scanner before signing macros.
- **Medium** A warning is displayed whenever a file is opened from a source that is not on your list of trusted sources (described below). You can choose whether to open the database.
- **Low** If you are sure that all the files and add-ins you open are safe, you can select this option. At this security level, macros are always enabled when you open files.

By default, the security level is set to Medium. If the security level is set to

Medium or High, you can maintain a list of trusted macro sources. When you open a file or load an add-in that contains macros developed by any of these sources, the macros are automatically enabled.

[Digital signatures](#)

A digital signature on a macro is like a wax seal on an envelope— it confirms that the macro originated from the developer who signed it and that the macro has not been altered.

When you open a file or load an add-in that contains a digitally signed macro, the digital signature appears on your computer as a certificate. The certificate names the source of the macro and provides additional information about the identity and integrity of that source. A digital signature does not necessarily guarantee the safety of a macro, so you must decide whether you trust a macro that has been digitally signed. For example, you might trust macros signed by someone you know or by a well-established company. If you are unsure about a file or an add-in that contains digitally signed macros, carefully examine the certificate before you open the file. If you know that you can always trust macros from a particular publisher or developer, you can add that publisher or developer to the list of trusted sources when you open the file or load the add-in.

You can digitally sign macros from within the [Visual Basic Editor](#).

[List of trusted publishers](#)

When you open a file that includes signed macros, you are prompted whether you want to trust all macros originating from that publisher. If you select this option, you add the certificate's owner to your list of trusted publishers. Before you decide to do this, you should review the details of the digital certificate — for example, look at the **Issued to** and **Issued by** fields to determine whether you trust its publisher, and look at the **Valid from** field to determine if the certificate is current. The certificate may also include details such as the e-mail name or Web site of the person who obtained the certificate.

Once you add a person (or corporation) to your list of trusted publishers, Office will enable macros signed by this trusted publisher without showing you a security warning. You can also remove entries from the list of trusted publishers.

Note A source added to the list of trusted publishers will also be trusted in Microsoft Internet Explorer.

[Security warning information](#)

[Notes about the security warning](#)

- The database might not be from a trusted source. If the security level is set to **Medium** or **High**, and you open a database that contains digitally signed macros that are not from a trusted source, you receive a security warning. If you are sure you trust the macro developer, add the name to the list of trusted publishers on the **Trusted Publishers** tab in the **Security** dialog box (on the **Tools** menu, point to **Macro**, and click **Security**).
- The database might contain a macro virus. If you don't expect the database to contain a macro, Visual Basic for Applications (VBA) code, a form that contains an ActiveX control, a report that contains an ActiveX control, action query, data-definition query, pass-through query, or update query, you might have a virus that is adding a macro virus to the database. Check your computer for viruses, and try to get an uninfected copy of the database from the source.
- The database contains legitimate macros, Visual Basic for Applications (VBA) code, forms that contain an ActiveX control, reports that contain an ActiveX control, action queries, data-definition queries, pass-through queries, or update queries. If you know these are legitimate and safe, you might want to digitally sign the database and then add your name to the list of trusted publishers on the **Trusted Publishers** tab in the **Security** dialog box (on the **Tools** menu, point to **Macro**, and click **Security**).

[Preventing macro security warnings](#)

To stop displaying macro warnings, change the security level to **Low** on the **Security Level** tab (on the **Tools** menu, point to **Macro**, and click **Security**). To avoid a macro virus infection on your computer, change the security level to **Low** only if you are running specialized antivirus software that can check documents for macro viruses, and you are sure all the macros you use are from trusted sources.

[When the Security Warning dialog box tells you the source has not been](#)

[authenticated](#)

This warning appears in the **Security Warning** dialog box if the security level is set to **High** or **Medium** on the **Security Level** tab (on the **Tools** menu, point to **Macro**, and click **Security**), and you open a database that contains digitally signed macros but the digital certificate has not been authenticated. For example, if the database developer has created his or her own digital certificate, you receive this warning.

[Show All](#)

Macro security levels

The following information summarizes how [macro](#) virus protection works under each setting on the **Security Level** tab in the **Security** dialog box (**Tools** menu, **Macro** submenu) under different conditions. Under all settings, if antivirus software that works with Microsoft Office 2003 is installed and the file contains macros, the file is scanned for known [viruses](#) before it is opened.

Security Because macros can contain viruses, be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; clear the **Trust all installed add-ins and templates** check box; use digital signatures; maintain a list of trusted publishers.

[Very High](#)

This setting is not available in the **Security** dialog box in Microsoft Office Access 2003. It is possible, however, to use system policies to set the security level in Access to **Very High**. When the security level is set to **Very High**, Access cannot open any [Access database](#) or [Access project](#) files

[High](#)

Unsigned macros

Macros are automatically disabled, and the file is not opened.

Signed macros

The source of the macro and the status of the [signature](#) determine how signed macros are handled.

[A trusted source. Signature is valid.](#)

Macros are automatically enabled, and the file is opened.

[An unknown author. Signature is valid.](#)

A dialog box is displayed with information about the [certificate](#). The file can be opened only if the user chooses to trust the author and certification authority. A network administrator can lock the list of trusted sources and prevent the user from adding the developer to the list and opening the file.

[Any author. Signature is invalid, possibly because of a virus.](#)

User is warned of a possible virus, and the file is not opened.

[Any author. Signature validation is not possible because public key is missing or incompatible encryption methods were used.](#)

User is warned that signature validation is not possible, and the file is not opened.

[Any author. The signature was made after the certificate had expired or had been revoked.](#)

User is warned that the signature has expired or been revoked, and the file is not opened.

[Medium](#)

Unsigned macros

User is prompted to open the database or cancel the operation. If the user chooses to cancel the operation, the database is not opened.

Signed macros

The source of the macro and the status of the [signature](#) determine how signed macros are handled.

[A trusted source. Signature is valid.](#)

Macros are automatically enabled, and the file is opened.

[An unknown author. Signature is valid.](#)

A dialog box is displayed with information about the certificate. The user is prompted to open the file or cancel the operation. The user can choose to trust the developer and certification authority. If the user chooses to cancel the operation, then the file is not opened.

[Any author. Signature is invalid, possibly because of a virus.](#)

User is warned of a possible virus, and the file is not opened.

[Any author. Signature validation is not possible because public key is missing or incompatible encryption methods were used.](#)

User is warned that signature validation is not possible. User is prompted to open the file or cancel the operation. If the user chooses to cancel the operation, then the file is not opened.

[Any author. The signature was made after the certificate had expired or had been revoked, and then the file is not opened.](#)

User is warned that the signature has expired or been revoked. User is prompted to open the file or cancel the operation. If the user chooses to cancel the operation, then the file is not opened.

[Low](#)

When security is set to low, all macros are treated equally regardless of origin or certificate status. With low security, you receive no prompt or signature validation and macros are automatically enabled. Use this setting only if you are certain that all macros in your files are from trusted sources.

Change the security level for macro virus protection

Security Because macros can contain viruses, be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; clear the **Trust all installed add-ins and templates** check box; use digital signatures; maintain a list of trusted publishers.

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Security Level** tab, and then select the security level you want to use.

[Show All](#)

Modify the list of trusted publishers for macros

Security Because macros can contain viruses, be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; clear the **Trust all installed add-ins and templates** check box; use digital signatures; maintain a list of trusted publishers.

To perform these procedures, you must have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later installed on your computer.

Do one of the following:

[Add a macro developer to the list of trusted publishers](#)

1. If you haven't already done so, set the [macro](#) security level to Medium or High.

[How?](#)

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Security Level** tab, and then select the security level you want to use.
2. Open the file or load the [add-in](#) that contains macros certified by the macro developer that you want to add to the list.
3. In the **Security Warning** box, select the **Always trust macros from this publisher** check box.

Note If the **Security Warning** box does not display the **Always trust macros from this publisher** check box, the macros are not [digitally signed](#). You cannot add this macro developer to the list of trusted sources without a digital signature.

[Remove a macro developer from the list of trusted publishers](#)

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Trusted Publishers** tab.
3. Click the source you want to remove from the list.
4. Click **Remove**.

[Show All](#)

Troubleshoot macro security

[I don't want to see the warning about macros.](#)

The macro might not be from a trusted source If the security level is set to Medium or High and you open a file or load an [add-in program](#) that contains [digitally signed](#) macros that are not from a trusted source, you receive a macro warning. If you are sure that you trust the macro developer, add the name to the list of trusted sources.

The macro might contain a virus If you don't expect a file to contain a macro, you might have a virus that is adding a macro virus to the file. Check your computer for viruses and then try to get an uninfected copy of the file from the source.

The macro might be in a temporary file that you downloaded from a Web site If your virus scanner tells you that it has removed a virus in this file every time you open it, you are removing the virus from the temporary file only and not from the original file on the Web server. Check your computer for viruses and then notify the source about problems with the original file.

The database may contain legitimate macros Microsoft Office Access cannot distinguish between safe and unsafe macros. If you know that these are legitimate and safe macros, you might want to digitally sign the macros and add your name to the list of trusted sources.

[The **Trust access to Visual Basic Project** setting in the **Security dialog box** is unavailable.](#)

The **Trust access to Visual Basic Project** setting is not available in Access. You cannot block programmatic access to the Visual Basic Project.

[Show All](#)

Troubleshoot macro security and warnings

Restricted Access

[I can't change the macro security level.](#)

Your network administrator might have enforced a security level for your workgroup or corporation to ensure that you use only macros that have been determined to be virus-free. For more information, see your network administrator.

Macro Warnings

[I'm told that my database can't be opened because Access security settings restrict access to the file.](#)

You receive this message for one of the following reasons:

- The macro security level is set to Very High.
- The macro security level is set to High.
- The macro security level is set to High and the database is not signed.
- Access cannot determine the status of the database's [signature](#).

[I don't want to be warned about any macros.](#)

To avoid macro warnings, attach a digital signature to each macro project and add that signature to your list of trusted sources. If the macros are already signed and if you are willing to trust all macros signed with that certificate, add the signer to your list of trusted sources. This will stop macro warnings when your Security setting is set to High or Medium.

Note The presence of a certificate does not guarantee that a macro is safe. Always review the details of the certificate—for example, look at the **Issued to** and **Issued by** fields to determine whether you trust its publisher, and look at the **Valid from** field to determine if the certificate is current.

A less secure option is to change the security level to Low. When your macro security level is set to Low, Microsoft Access will not provide warnings about macros. To reduce your risk of getting a macro virus infection on your computer, run specialized antivirus software that is up to date and that can check files and add-ins for macro viruses and use macros only from trusted sources.

[I keep getting a warning about macros.](#)

- **The database might not be from a trusted source** If the security level is set to Medium or High, and you open a database that contains digitally signed macros that are not from a trusted source, you receive a security warning. If you are sure you trust the macro developer, add the name to the

list of trusted publishers on the **Trusted Publishers** tab in the **Security** dialog box (on the **Tools** menu, point to **Macro**, and click **Security**).

- **The database might contain a macro virus** If you don't expect the database to contain a [macro](#), Visual Basic for Applications (VBA) code, a [form](#) that contains an [ActiveX control](#), a [report](#) that contains an [ActiveX control](#), [action query](#), [data-definition query](#), [pass-through query](#), or [update query](#) you might have a virus that is adding a macro virus to the database. Check your computer for viruses, and try to get an uninfected copy of the database from the source.
- **The database contains legitimate macros** If you know these are legitimate and safe, you might want to digitally sign the database and then add your name to the list of trusted publishers on the **Trusted Publishers** tab in the **Security** dialog box (on the **Tools** menu, point to **Macro**, and click **Security**).

[Show All](#)

Warn about installed templates and add-ins

Security Because macros can contain viruses, be careful about running them. Take the following precautions: run up-to-date antivirus software on your computer; set your macro security level to high; clear the **Trust all installed add-ins and templates** check box; use digital signatures; maintain a list of trusted publishers.

Depending on your [macro](#) security setting, when you open a macro, you will receive a warning and the macro may be disabled for installed [templates](#) and [add-ins](#) (including wizards).

1. On the **Tools** menu, point to **Macro**, and then click **Security**.
2. Click the **Trusted Publishers** tab.
3. Clear the **Trust all installed add-ins and templates** check box.

Note Clearing the **Trust all installed add-ins and templates** check box will prevent the wizards installed by Access from functioning if macro security is set to High. Users will be prompted to enable macros if macro security is set to Medium.

[Show All](#)

About the types of Web pages Access creates

Using Microsoft Access, you can create different types of Web pages. To manipulate the data directly in your database, use a data access page; to see up-to-date, read-only data, consider server-generated [ASP](#) or [IDC/HTX](#) files; to see a snapshot of your data, use a static [HTML](#) file format. To make sure that the appearance of your Web pages is consistent, you can use HTML template files.

[Data access pages](#)

A data access page is a Web page that is connected directly to the data in your database. Use these pages to view, edit, update, delete, filter, group, and sort live data from either an Access database or a [Microsoft SQL Server database](#), in Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later. A page can also contain components such as a [spreadsheet](#), a [PivotTable list](#), or a [chart](#).

You create a data access page as a [database object](#) that contains a shortcut to the location of the page's corresponding HTML file.

Data access pages can also be saved as HTML Application (HTA) files. HTA files allow you to use Dynamic HTML ([DHTML](#)) technology "out of the browser" to write stand-alone applications. With HTA, you can write an application the same way you write a Web page with DHTML and script, or you can take content originally authored for the Web and turn it into an application. "Out of the browser" means that HTAs differ from Web pages in two important ways:

- Your application is written completely in DHTML but runs in its own window without the browser menus and toolbars. This means your application fully defines the user interface.
- Your application is fully trusted and free from the restrictions placed on Web pages for security reasons. Unlike Web pages, which run when visited, users will need to trust your HTA; however, once installed and run, your HTA can potentially do anything any program can.

More information about HTA files can be found on Microsoft's [MSDN](#) site on the Web.

Making your data access pages available

To make your pages available on the [World Wide Web](#), you publish the pages to [Web folders](#) or to a Web server. You must also make the Access database or SQL Server database available to users of the page.

Internet Explorer needs to download the page only once from the Web server to let you view and interact with the data on the page. Because a page uses DHTML, access to the database is generally very efficient in a client/server environment.

The Access or SQL Server database that is the [OLE DB](#) data source for a data access page must be on a shared server or computer in order for users to view and interact with the page in a Web browser or in a mail message.

It's a good idea to place the database on a shared server or computer before you create your page. If you move the database that the page is connected to after you create the page, you must change the OLE DB data source connection of the page in the **Connection** dialog box of the data access page.

Although the data source of a page can only be an Access or SQL Server database, you can make other data formats accessible from a page by using linked tables from an Access database.

To further help protect your data access pages, you must put the Access or SQL Server database on the same Web server as the data access page.

Important Make sure you use a universal naming convention ([UNC](#)) path, instead of relying on the drive letter of a mapped network drive in Windows Explorer when you are setting up the data source to the page. A drive letter can vary on a computer or may not always be defined, whereas a UNC path is a reliable and consistent way for the page to locate the data source.

Supporting files and Web links

When you create a data access page, Access manages the related files and plans the links and hyperlinks so that the images appear and the links work when the pages are placed on the final Web server.

When you save your file as a data access page in [Design view](#), all supporting files— such as bullets, background textures, and graphics— are by default organized in a supporting folder. If you move or copy your data access page to another location, you must also move the supporting folder so that you maintain all links to your page.

For example, suppose you have a page called Page1.htm. It includes bullets, which are stored in a supporting folder called Page1_files. If you move Page1.htm, you must also move the supporting folder (Page1_files) to the new location.

By default, the name of the supporting folder is the name of the data access page plus an underscore (_), a period (.), or a hyphen (-), and the word "files." The word "files" will appear in the language of the version of Microsoft Office used to save the file as a data access page. For example, if you use the Dutch language version of Office to save a file called Page1 as a data access page, the default name of the supporting folder would be **Page1_bestanden**.

After you have published your data access page to the [Internet](#) or an [intranet](#), you should test it in a Web browser. Make sure that all the graphics are displayed and that hyperlinks go to the correct destinations.

[Dynamic \(server-generated\) HTML](#)

You can create dynamic (server-generated) HTML files, either ASP or IDC/HTX, from tables, queries, and forms. Server-generated HTML files are displayed in a table format in a Web browser. Use server-generated HTML files when you want to use any Web browser. Server-generated HTML files are also useful if you need to see live data in a table connected to an [ODBC data source](#), but you only need read-only access to the data.

Once you have output a datasheet to ASP or IDC/HTX format, you need to

publish the files to make them available on the World Wide Web.

Each time a user opens or refreshes an ASP or HTX file from a Web browser, the Web server dynamically creates an HTML file, and then sends that HTML file to the Web browser.

If you create a server-generated HTML file from a datasheet that contains a [parameter query](#), Access simulates the **Enter Parameter Value** dialog box by creating an additional HTML parameter page that contains an HTML form text box control for entering the parameter value, and a button for running the query. You must explicitly define the parameter data types when you create the parameter query. You must also display this HTML parameter Web page before you display the datasheet Web page. During the export operation, Microsoft Access executes the query and displays the **Enter Parameter Value** dialog box, but you can just click **OK** because the input is ignored.

Table or datasheet fields with a Hyperlink data type are output in HTML files as strings by using the following hyperlink address format: *displaytext#address#subaddress#screentip*. Although you cannot click the string in a Web browser to follow the hyperlink address, you can copy and paste the address portion into the address bar of the Web browser.

Differences between ASP files and IDC/HTX files

You can save table, query, or form datasheets as IDC/HTX files that generate HTML documents by querying your database for current data.

You can save forms as ASP files that emulate most of the functionality of the original form and display data from your database on a Web server. Most of the controls on your form are saved as [ActiveX](#) controls that perform the same or similar functions. None of the Visual Basic code behind your forms or controls is saved or run when the ASP file is created or activated. You can also save tables, queries, and form datasheets as ASP files that display the current data from your database.

How your Web server processes IDC/HTX files

When you output a table, query, or form as IDC/HTX files, instead of as an .html file, Access creates an HTML extension file (.htx), and an Internet Database Connector file (.idc). Internet Database Connector (httpodbc.dll) is a component of Microsoft Internet Information Services (IIS).

The .idc file contains a query in the form of an SQL statement and information that IIS uses to connect to an ODBC data source such as an Access or SQL Server database. The connection information includes the data source name and often a user name and password.

The .htx file is an HTML file that contains formatting information and placeholders indicating where to insert the values returned from the query in the .idc file.

After you publish your IDC/HTX files to a Web server, IIS can open the database (by using the appropriate ODBC driver and the .idc file connection information), run the query in the .idc file to access the data, merge the results and the .htx file into one .html file, and send the dynamically created .html file to the Web browser as a Web page.

How your Web server processes ASP files

On the other hand, when you output a table, query, or form as an ASP file, instead of as an .html file, Access creates a Microsoft Active Server Page (.asp) file. Active Server is a component of Microsoft Internet Information Server 3.0 or later.

The .asp file contains HTML tags interspersed with one or more queries in the form of SQL statements, template directives, and Microsoft Visual Basic Scripting Edition (VBScript) code containing references to ActiveX server controls. The .asp file also contains ODBC connection information to connect to an ODBC data source; in this case, an Access or SQL Server database. The connection information includes the data source name and often a user name and password.

After you publish your ASP files to a Web server, IIS can run the VBScript code, call the ActiveX server controls, open the database (by using the appropriate ODBC driver and the .asp file connection information), run the queries in the .asp file to access the data, merge the results and HTML tags in the .asp file into one .html file, and then send the dynamically created .html file to the Web browser as a Web page.

[Static HTML](#)

You can create static HTML files from tables, queries, forms, and reports. In a Web browser, reports are displayed in a report format. Tables, queries, and forms are displayed in a datasheet format. Use static HTML files when you want to use any Web browser that supports HTML version 3.2 or later and if your data does not change frequently.

To make your static HTML files available on the World Wide Web, you publish the files to Web folders or a Web server.

When you access the data through a Web browser, the browser needs to download the static HTML file only once from the Web server to let you view the data. There is no [ODBC data source](#) connected to the static HTML file, and if your data changes, you must export your files again to be able to view new data in a Web browser.

[HTML template files](#)

You can use one or more HTML template files to enhance the appearance, consistency, and navigation of your static HTML and server-generated HTML files. For example, you might want to include a company logo in the [header](#) section and your own navigation buttons in the [footer](#) section of a static HTML report.

An HTML template is a text file that includes HTML tags and tokens that are unique to Access. These tokens indicate where to insert output and other information in the static HTML or server-generated HTML files.

When you output a table, query, form, or report using the **Export** command on the **File** menu and you specify an HTML template file in the **Output Options** dialog box, Access merges the HTML template file with the .html, .asp, and .htx output files by replacing the tokens (such as <!--AccessTemplate_Title-->) with the following:

<!--AccessTemplate_Title-->

Becomes the name of the table, query, form, or report placed in the title bar of the Web browser

<!--AccessTemplate_Body-->

The output of the table, query, form, or report

<!--AccessTemplate_FirstPage-->

An HTML anchor tag that links to the first page of a report

<!--AccessTemplate_PreviousPage-->

An HTML anchor tag that links to the page previous to the current report page

<!--AccessTemplate_NextPage-->

An HTML anchor tag that links to the next page after the current report page

<!--AccessTemplate_LastPage-->

An HTML anchor tag that links to the last page of a report

<!--AccessTemplate_PageNumber-->

The current page number

HTML template file example

```
<HTML>
```

```
<!--The following token places the object name in the title bar of the Web browser.-->
```

```
<TITLE><!--AccessTemplate_Title--></TITLE>
```

```
<!--The following HTML tag creates a different background color than the Web browser default.-->
```

```
<BODY BACKGROUND = "gray.jpg">
```

```
<!--The following token places all object output inside the <BODY> tag.-->
```

```
<!--AccessTemplate_Body-->
```

```
</BODY>
```

```
<BR><BR>
```

```
<!--The following four tokens create four navigation text buttons that jump to the first, previous, next, and last pages of a report.-->
```

```
<A HREF = "<!--AccessTemplate_FirstPage-->">First</A>
```

```
<A HREF = "<!--AccessTemplate_PreviousPage-->">Previous</A>
```

```
<A HREF = "<!--AccessTemplate_NextPage-->">Next</A>
```

```
<A HREF = "<!--AccessTemplate_LastPage-->">Last</A>
```

```
<!--The following token inserts the text "Page n" , where n is the current report page number.-->
```

```
<P ALIGN = CENTER>Page <!--AccessTemplate_PageNumber-->.</P>
```

```
<!--The following HTML tag adds a company logo to the bottom of the Web
```

page.-->

</HTML>

[Show All](#)

About where to save on the Web

The steps that you take to save your [data access pages](#) to the Web depend on where you want to share them.



[Making your data access pages available on an intranet](#)

If your company uses an [intranet](#) that is based on [Internet](#) protocols, you can use Web Folders to save copies of your data access pages to a [Web server](#). You must have access to a Web server before you can copy or manage folders and files on a Web server. To get the [URL](#) of a Web server that you can save files to, see your system administrator.

- If your Web server supports Web folders, you can use Web folders to save copies of your Web pages to the Web server. Before you can copy or manage folders and files on a Web server, see your system administrator or Internet service provider (ISP) to get access to and the URL of a Web server you can save files to.
- If your Web server supports [File Transfer Protocol \(FTP\)](#) you can save a data access page to an FTP site. See your system administrator or Internet service provider to get access to an FTP site you can save files to.
- If you have questions about saving your data access pages to the intranet, see your system administrator.



[Making your data access page available on the Internet](#)

To make your data access pages available on the [Internet](#), you need to either locate an Internet service provider that allocates space for data access pages, or install Web server software on your computer. Some factors to consider in setting up your computer as a Web server are your computer's speed and availability. If you don't want to leave your computer on all day or most of the day, you might not want to set up your computer as a Web server.

Here are some options for saving your Web pages on the Internet.

- If your Web server supports Web folders, you can use Web folders to save

copies of your Web pages to the Web server. Before you can copy or manage folders and files on a Web server, see your system administrator or Internet service provider (ISP) to get access to and the [URL](#) of a Web server you can save files to.

- If your Web server supports [File Transfer Protocol \(FTP\)](#), you can save a data access page to an FTP site. See your system administrator or ISP to get access to a location you can save files to.
- If you have questions about saving your data access pages to the Internet, see your system administrator or ISP.

[Managing data access pages on an intranet or the Internet](#)

If you are working with an Internet service provider (ISP) or your organization's Web administrator, you should ask how the data access pages, graphics files, and other files should be structured on the server. For example, find out whether you need to create separate folders for bullets and pictures, or whether you need to store all the files in one location. If you plan to use forms or image maps, ask about any limitations on using these items, as they require additional server support.

[Show All](#)

Turn off automatic completion of file names and URLs

If you have Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later, you can turn off the feature that automatically suggests a match for file names and [URLs](#) as you type them in the **Insert Hyperlink** dialog boxes in Microsoft Office programs, and in the **Address** box in Internet Explorer.

1. Start or switch to Internet Explorer.
2. Click **Internet Options** on the **Tools** menu, and on the **Advanced** tab, clear the **Use inline AutoComplete** check box.

[Show All](#)

About online meetings

The integration of Microsoft Windows NetMeeting with Microsoft Office allows you to share and make changes to a document with people at different sites in [real time](#). When you start an online meeting from within an Office application, NetMeeting automatically starts in the background and allows you to begin sharing the contents of your document. Once you are in an online meeting, you can share applications and documents, send text messages in [Chat](#), transfer files, and work on the [Whiteboard](#).

You can either host an online meeting or be invited to participate in one. The host of the online meeting and all participants must have NetMeeting installed on their computers, but the host is the only person who needs to have the shared document and its application installed.

[Hosting an online meeting](#)

Start the meeting You can either schedule the meeting in advance by using Microsoft Outlook (see Microsoft Outlook Help) or start an impromptu online meeting from within the [Microsoft Access database](#), [Microsoft Access project](#), or [data access page](#) you want to share. If the participants are available and they decide to accept your invitation, the online meeting begins. Also, the host should confirm that all participants know which server will be used for the meeting and can access it.

Control the Access database, Access project, or data access page At the start of an online meeting, the host is the only person who has control of the Access database, Access project, or data access page, although all participants can see it on their screens. As the host, you can allow participants to make changes to the shared document by turning on collaboration, and you can turn off this collaboration at any time. When collaboration is turned off, the other participants can no longer make changes, but they will be able to watch you work. When collaboration is turned on, each person in the online meeting can take turns editing and controlling the Access [data file](#). When someone else is in control, you will not have the use of your cursor— not only in the Access data file, but also for any other purpose. The initials of the person in control appear

next to the pointer.

Add participants to a meeting in progress You can invite participants to join an online meeting any time you have control of the cursor during the meeting. Participants whom you invite to an impromptu online meeting must have NetMeeting running on their computers for them to receive your online meeting invitation.

[Participating in an online meeting](#)

When you're invited to participate in an online meeting, you'll receive a message requesting you to join an upcoming or ongoing online meeting at a specified time and date. When you accept a call to join an online meeting, the **Online Meeting** toolbar and the [Microsoft Access database](#), [Microsoft Access project](#), or [data access page](#) that is being shared appear on your screen.

[Collaborating in an online meeting](#)

In an online meeting, you can share applications and documents, send text messages in Chat, transfer files, and work on the Whiteboard. By collaborating, participants can take control of the shared document to review and edit its contents. During an online meeting, only one person at a time can control the [Access file](#), but multiple users can work in Chat or the Whiteboard simultaneously if collaboration is turned off. The host of the online meeting is the only person who needs to have the shared document and its application installed.

[Show All](#)

Control an Access database, Access project, or data access page in an online meeting

The host controls the [Microsoft Access database](#), [Microsoft Access project](#), or [data access page](#) when the meeting starts. The host can allow collaboration by the participants and can turn off collaboration at any time. When collaboration is turned on, any participant can control the Access database, Access project, or data access page, but only one person at a time can have control. When collaboration is turned off or on, multiple users can work simultaneously in [Chat](#) or on the [Whiteboard](#).

Control the Access database, Access project, or data access page as the host

- To turn on collaboration, click **Allow others to edit**  on the **Online Meeting** toolbar.
- To take back control of the Access database, Access project, or data access page when collaboration is turned on, click anywhere in the [Access data file](#).
- To turn off collaboration, click **Allow others to edit** again if you have control, or press ESC if you don't have control.

Control the Access database, Access project, or data access page as a participant

To take control of a shared Access database, Access project, or data access page the first time, double-click anywhere in the Access data file. Thereafter, you can gain control by clicking anywhere in the Access data file.

Only the host of the meeting can use the **Online Meeting** toolbar. If you want more control of Microsoft NetMeeting features during an online meeting you must use the NetMeeting program directly by clicking **Microsoft NetMeeting** on the taskbar.

Note The initials of the person in control appear next to the pointer.

[Show All](#)

Use the Whiteboard in an online meeting

Only the host of the online meeting can open the Whiteboard window from within Microsoft Office.

If participants want to open the Whiteboard window, they must use Microsoft NetMeeting directly.

1. Click **Microsoft NetMeeting** on the taskbar.
2. On the **Online Meeting** toolbar, click **Display Whiteboard** .

Note In order for participants to draw or type simultaneously on the Whiteboard, the host must turn off collaboration; if you have control of the [Microsoft Access database](#), [Microsoft Access project](#), or [data access page](#), click **Allow others to edit**  on the **Online Meeting** toolbar. If you don't have control, press ESC to bring the focus to the database, Access project, or data access page.

[Show All](#)

Troubleshoot online meetings

[I can't print or save an Access file or data access page opened from an online meeting to my local printer or hard disk.](#)

When you use a Microsoft Access file or [data access page](#) during an online meeting, you can print or save the file only to the computer that hosted the online meeting. When you click **Print**, you print the file on the printer selected by the person hosting the online meeting. When you click **Save**, you save the file to the host's hard disk or server, not to your own machine.

If you want a copy of the file you're collaborating on, ask the host of the online meeting to send you a copy of the file through e-mail, or copy the document contents and paste the contents to a file on your computer.

[Other people in the online meeting see a crosshatch pattern on the screen.](#)

Make sure no windows are covering your [Microsoft Access database](#), [Microsoft Access project](#), or [data access page](#).

- Quit programs you are not using.
- Minimize other windows on your desktop.

[How?](#)

Right-click the background of the Windows taskbar, and then click **Minimize All Windows** on the [shortcut menu](#).

[I don't see some items that other people in the online meeting see.](#)

Make sure your screen resolution settings match the settings of the other online meeting participants. See Windows Help for more information about setting your screen resolution.

[The name of the person I want to invite to an online meeting is not in the list.](#)

Connect to another server

- In the **Place a Call** dialog box, select another directory in the **Directory** list.

Send an e-mail message or telephone the other person

- Let the person know that you want to have an online meeting. In the **Find Someone** dialog box, you need to select another directory in the **Directory** list, or simply notify the person you want to invite via e-mail or telephone.

[Which NetMeeting directory server should I connect to?](#)

The first time you start an online meeting, you'll be prompted to select a [directory server](#) to connect to. You can select any of the directory servers from the list in the **Server name** box under **Directory**. If you're a corporate user, ask your system administrator for the name of a directory server you can connect to.

[Show All](#)

About managing Web files and links

Some of the content in this topic may not be applicable to some languages.

When you create Web pages, your Microsoft Office application manages the related files and inserts the links and [hyperlinks](#) so that the images appear and the links work when the pages are placed on the [Web server](#).



[Managing links](#)

When you create hyperlinks in your Office files, you have the option of making the links [relative](#) or [absolute](#). In most cases, it's best to use relative links for any files that will be published on the same server - such as bullets, navigational buttons, background textures, graphics, and pages within the same Web. Using relative links makes it easier to move supporting files from one location to another

Hyperlinks to files on other Web servers, for example, a list of your favorite places on the Web, must be set up as absolute links.

If you use the **Save as Web page** option from an Office application to save your Web pages to a different server, any relative links to files that aren't being copied to the other server are converted to absolute links.



[Managing and naming supporting files](#)

By default, when you save your file as a Web page, all supporting files— such as bullets, background textures, and graphics— are organized in a supporting folder. If you move or copy your Web page to another location, you must also move the supporting folder so that you maintain all links to your Web page.

For example, suppose you have a Web page named Page1.htm. It includes bullets, which are stored in a supporting folder named Page1_files. The relative paths for the image files are \Page1_files\image001.gif and \Page1_files\image002.gif. If you move Page1.htm, you must also move the supporting folder (Page1_files) to the new location.

By default, the name of the supporting folder is the name of the Web page plus an underscore (_), a period (.), or a hyphen (-), and the word "files." For some language versions of Office, the word "files" is translated. For example, suppose you use the Dutch language version of Office 2000 to save a file named Page1 as a Web page. The default name of the supporting folder would be **Page1_bestanden**.

[Default names for Web page supporting folders](#)

By default, when you save your file as a Web page, all supporting files— such as bullets, background textures, and graphics— are organized in a supporting folder. The name of the supporting folder is the name of the Web page plus an underscore (_), a period (.), or a hyphen (-), and the word "files." The word "files" will appear in the language of the version of Microsoft Office that was used to save the file as a Web page. For example, suppose you use the Dutch language version of Office to save a file named Page1 as a Web page. The default name of the supporting folder would be **Page1_bestanden**.

Note If you save your Web page with a short file name (maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the Web page without the word "files."

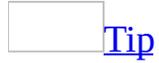
[The default name for the Web page supporting folder in each language version of Office](#)

Language	Default name for Web page supporting folder
Arabic	.files
Basque	_fitxategiak
Brazilian	_arquivos
Bulgarian	.files
Catalan	_fitxers
Chinese (Simplified)	.files
Chinese (Traditional)	.files
Croatian	_datoteke
Czech	_soubory

Danish	-filer
Dutch	_bestanden
English	_files
Estonian	_failid
Finnish	_tiedostot
French	_fichiers
German	-Dateien
Greek	.files
Hebrew	.files
Hungarian	_elemei
Italian	-file
Japanese	.files
Korean	.files
Latvian	_fails
Lithuanian	_bylos
Norwegian	-filer
Polish	_pliki
Portuguese	_ficheiros
Romanian	.files
Russian	.files
Serbian (Cyrillic)	.files
Serbian (Latin)	_fajlovi
Slovakian	.files
Slovenian	_datoteke
Spanish	_archivos
Swedish	-filer
Thai	.files
Turkish	_dosyalar
Ukrainian	.files
Vietnamese	.files

[Testing links](#)

After you have published your Web pages to the Internet or an intranet, you should test them in a Web browser. Make sure that all the graphics are displayed and that hyperlinks go to the correct destinations.



Tip

You can save your file as a [Single File Web Page](#) (MHTML) file so that all the text and graphics are stored in a single file. You can then move or send the MHTML file as an e-mail attachment without worrying about broken links. You can view MHTML files with Internet Explorer 4.0 and later.

[Show All](#)

Troubleshoot Web page options

Some of the content in this topic may not be applicable to some languages.

Working with files and links

[I don't want to keep my supporting files in a folder](#)

By default, when you save a Web page to a [Web server](#), all supporting files — such as bullets, background textures, and graphics— are stored in a separate folder. If you do not want to use a separate folder, do one of the following:

[Save your Web page as a Single File Web Page](#)

1. Click **File**, and then click **Save as Web Page**.
2. In the **Save as type** list, select **Single File Web Page**.
3. In the **File name** box, type a name for the file, and then click **Save**.

[Change the Web options settings so that the supporting files are not saved in a folder](#)

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab, and then click **Web Options**.
3. Click the **Files** tab.
4. Under **File names and locations**, clear the **Organize supporting files in a folder** check box.

[I saved my Web page to a file server, and now some people can't find or view it](#)

If you use a long file name to save your Web page to a file server, site visitors running Windows 3.1 won't be able to find or open your Web page, because Windows 3.1 recognizes and supports only short file names (maximum of eight characters, plus a three-character file extension). To always save files with short file names, do the following:

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab, and then click **Web Options**.
3. Click the **Files** tab.
4. Under **File names and locations**, clear the **Use long file names whenever possible** check box.

[I moved my Web page to another location and now some of the links are broken](#)

By default, when you save your file as a Web page, all supporting files— such as bullets, background textures, and graphics— are organized in a supporting folder. If you move or copy your Web page to another location, you must also move the supporting folder so that you maintain all links to your Web page.

[I can't automatically create a backup copy of my Web page](#)

When you save a Web page, a backup copy is not automatically created. To save a copy of a Web page:

1. On the **File** menu, click **Save As Web Page**.
2. In the **File name** box, enter a new name for the file, and then click **Save**.

[The name of a supporting folder for a Web page is in another language](#)

By default, when you save your file as a Web page, all supporting files— such as bullets, background textures, and graphics— are organized in a supporting folder. The name of the supporting folder is the name of the Web page plus an underscore (_), a period (.), or a hyphen (-), and the word "files." For some language versions of Office, the word "files" is translated. For example, suppose you use the Dutch language version of Office to save a file named Page1 as a Web page. The default name of the supporting folder would be **Page1_bestanden**.

[Default name for supporting folders](#)

By default, when you save your file as a Web page, all supporting files— such as bullets, background textures, and graphics— are organized in a supporting folder. The name of the supporting folder is the name of the Web page plus an underscore (_), a period (.), or a hyphen (-), and the word "files." The word "files" will appear in the language of the version of Microsoft Office that was used to save the file as a Web page. For example, suppose you use the Dutch language version of Office to save a file named Page1 as a Web page. The default name of the supporting folder would be **Page1_bestanden**.

Note If you save your Web page with a short file name (maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the Web page without the word "files."

[The default name for the Web page supporting folder in each language version of Office](#)

Language	Default name for Web page supporting folder
Arabic	.files
Basque	_fitxategiak
Brazilian	_arquivos
Bulgarian	.files
Catalan	_fitxers
Chinese (Simplified)	.files
Chinese (Traditional)	.files
Croatian	_datoteke
Czech	_soubory
Danish	-filer
Dutch	_bestanden
English	_files
Estonian	_failid
Finnish	_tiedostot
French	_fichiers
German	-Dateien
Greek	.files
Hebrew	.files
Hungarian	_elemei
Italian	-file
Japanese	.files
Korean	.files
Latvian	_fails
Lithuanian	_bylos
Norwegian	-filer

Polish	_pliki
Portuguese	_ficheiros
Romanian	.files
Russian	.files
Serbian (Cyrillic)	.files
Serbian (Latin)	_fajlovi
Slovakian	.files
Slovenian	_datoteke
Spanish	_archivos
Swedish	-filer
Thai	.files
Turkish	_dosyalar
Ukrainian	.files
Vietnamese	.files

Note If you save your Web page with a short file name (maximum of eight characters, plus a three-character file extension) by clearing the **Use long file names whenever possible** check box in the **Web Options** dialog box, the supporting folder is the name of the Web page without the word "files."

[The HTML file I want to edit opens in the wrong Office application](#)

When you open a [Hypertext Markup Language \(HTML\)](#) or [Single File Web Page](#) file from the **Open** dialog box on the **File** menu, the file opens in the program the file was created in. For example, if you attempt to open an HTML format file in Microsoft PowerPoint that was created in Word, the file opens in Word instead. To open an HTML file in PowerPoint that was created in another Microsoft Office program, right-click the file in the **Open** dialog box, and then click **Open in Microsoft PowerPoint** on the shortcut menu.

Note For best results when you edit an HTML or Single File Web Page (MHTML) file, open it in the program the file was created in.

[My Web page has the wrong title](#)

The Web page title is the text that appears in the title bar when you view the Web

page in your [Web browser](#). If someone stores a link to your Web page, the title appears in that person's history list and favorites list.

1. Open an Office application such as Microsoft Word, Excel, or PowerPoint.
2. Click **File**, and then click **Open**.
3. Double-click the folder that contains the Web page with the title you want to change.
4. Double-click the Web page.
5. Click **File**, and then click **Save as Web Page**.
6. Click **Change Title**, and then type the text you want as your title in the **Page title** box.

Working with graphics

[I see a red X, a blank image, or no image at all on my Web page](#)

If you see a red X, a blank image, or no image at all where your graphics should be, check the following:

[Your links might not be working correctly](#)

A red X indicates a missing graphic. If you move, copy, or rename your Web page, any links to supporting files— such as bullets, background textures, and graphics— might be broken. To automatically make the paths for the linked pictures and hyperlinks [relative](#) when you save a Web page, do the following:

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab, and then click **Web Options**.
3. Click the **Files** tab.
4. Under **File names and locations**, select the **Update links on save** check box.

[Your browser might not recognize graphics in VML format](#)

If you save a file as a Web page and then view the Web page in a browser that supports graphics in [Vector Markup Language \(VML\)](#) format, such as Microsoft Internet Explorer 5.01 with Service Pack 2 (SP2) or later, make sure that VML is turned on in your Web option settings in order to see the graphics on your Web page. If you view the Web page in a browser that does not recognize graphics in VML format (such as Microsoft Internet Explorer 4.0 or earlier) and your graphics aren't showing, check to make sure that you have set the correct Web option by doing the following:

1. Open your Web page in the Office application that you used to create it.
2. On the **Tools** menu, click **Options**.
3. Click the **General** tab, and then click **Web Options**.
4. Click the **Browsers** tab.
5. Under **Options**, clear the **Rely on VML for displaying graphics in browsers** check box, and then save your Web page again.

[You might have the wrong browser settings](#)

Although all browsers support the same language (HTML), there are some differences among browsers. Some support different versions of HTML and different [HTML extensions](#). Custom options can be set in browsers, such as the default text and background colors, and whether or not graphics are displayed. Consider testing your page in different browsers— some of which are available for downloading on the Web— and with different monitor settings, to make sure that all of your page elements appear.

[You might have used the wrong picture format](#)

If you link a graphic to a Web page, and the graphic format isn't supported by your browser, your graphics won't be visible. Make sure that the linked picture is in a graphic format that is compatible with your browser. The .jpg and .gif graphic formats are compatible with most browsers.

[You might have changed a linked picture](#)

If you make a visual change to a linked picture, such as adding a shadow, border or outline, fill, or contrast or brightness adjustment, then your Office application will create another picture with those changes for displaying in a Web page. When you view the Web page in Microsoft Internet Explorer 4.0 or earlier, the picture will appear with the changes, but the link to the original picture will be broken. To fix the link:

1. Select the picture.
2. On the **Picture** toolbar, click **Reset Picture**.

[The graphics and text on my Web page look different on another computer](#)

If your Web page looks different from what you expect, check the following:

[Video resolution](#)

Graphics and text can wrap differently, depending on the video resolution (the size of the screen elements relative to the screen size). The screen size you

specify can affect the size and layout of images on a Web page and the image of gradient page backgrounds.

[Check the screen size settings on your computer](#)

1. In the Windows Control Panel, double-click the **Display**  icon.
2. Click the **Settings** tab.

Note Your monitor and video adapter determine your screen resolution options.

If your graphics and text look different from what you expect, change the screen size settings by doing the following:

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab, and then click **Web Options**.
3. Click the **Pictures** tab.
4. Under **Target monitor**, select a screen size in the **Screen size** box.

[Browser settings](#)

Although all Web browsers support the same language ([HTML](#)), there are some differences among browsers. Some support different versions of HTML and different [HTML extensions](#). In addition, users can set custom options in browsers, such as the default text and background colors, and whether or not graphics are displayed. Consider testing your page in different browsers— some of which are available for downloading on the Web— and with different monitor settings to make sure all of your page elements appear.

Specifying the language and fonts

[My Web page is displaying the wrong characters for a language](#)

If a Web page is [encoded](#) for a different language, an Office application will try to determine the language. If the Office application displays the wrong characters for that language when you open the page in a browser, you can select the language that you think the page is encoded in. Some languages have more than one encoding, so try each encoding until you can read the text.

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab, and then click **Web Options**.
3. Click the **Encoding** tab.
4. Under **Reload the current document as**, select the language you think the page is encoded in.

Working with different Web browsers

[My Web page doesn't look right in my Web browser](#)

1. On the **Tools** menu, click **Options**.
2. On the **General** tab, click **Web Options**.
3. On the **Browsers** tab, in the **People who view this Web page will be using** list, select the appropriate browser version.
4. In the **Options** list, select or clear the check box for any features you want to enable or disable.

Note When you choose a setting from the **People who view this Web page will be using** list, the default features for the selected browser version are set in the **Options** list.

[Show All](#)

About working with graphics on Web pages

Some of the content in this topic may not be applicable to some languages.

Graphic formats in Web pages

The first time you save your file in [HTML](#) format, all graphics are converted to [.gif](#), [.jpg](#), or [.png](#) format, all of which are image types supported on the World Wide Web. You can use graphics that are provided with Microsoft Office, such as clip art, AutoShapes, text boxes, and WordArt, or create your own graphics in [.gif](#) format. If you have [Internet](#) access, you can get more graphics from the Microsoft Web Art Resources Web page. When you reopen the Web page in your Office application, you can edit all graphics as you normally would.

Inserting graphics from a file

When you insert a graphic from a file, you can either insert the entire graphic or you can insert a link to the graphic at a [fixed location](#), such as another [Web server](#), or at a [relative file location](#). Linking a graphic to your Web page reduces the size of the page, makes it easier to update the graphic, and makes it easier to share the graphic between different Web pages. In Windows, if you move or copy a Web page to another location, you should make sure that you also move or copy any supporting files— such as graphics, background textures, and bullets— so that you maintain the graphic links to your Web page.



Tip

You can save your file as a [single file Web page](#) (MHTML) file so that all the text and graphics are stored in a single file. You can then move or send the MHTML file as an e-mail attachment without worrying about broken links. You can view MHTML files with Internet Explorer 4.0 and later.

Using art from Office application drawing tools

You can use [AutoShapes](#), text boxes, and WordArt effects in the files that you save to HTML from your Office applications. After you save your file to HTML format, you can open and view it in a Web browser, or you can open and edit it in any Office application. From your Office application, you can open and edit all drawings that you have saved to HTML format from any other Office application.

Using animated .gif images

You can insert animated .gif images into Office application files. When you save an animated .gif image in HTML format, the animation is preserved. You cannot perform certain editing operations— such as cropping, grouping, or changing the fill, border, or shadow— on an animated .gif image. Make these changes in a program that supports animated .gif editing, and then insert the animated .gif image in the file.

Aligning graphics

The **Drawing** and **Picture [toolbars](#)** in Office applications provide alignment commands that help you arrange your graphics. Selecting a picture automatically displays the **Picture** toolbar. To hide the **Picture** toolbar, right-click a picture, and then click **Hide Picture Toolbar** on the shortcut menu.

[Show All](#)

About creating and editing a Visual Basic script on a Web page

You can add Microsoft Visual Basic Script (VBScript) or [JScript](#) to your page in Microsoft Access.

[Scripting](#) is useful for custom Web-based solutions development. It provides a full object model for the Web browser and the objects on the current page. This makes it easy for you to write code that manipulates elements on the page without having to know [HTML](#) or how the objects are implemented. You can also create script code that is specifically designed to handle events that occur on objects (such as [ActiveX controls](#)) on your Web page.

To view the HTML code or edit a script for your Web page, open the [Microsoft Script Editor](#). In the Script Editor, you can add text, edit HTML tags, and edit any script code. You can also view your Web page as it would appear in a Web browser.

For more information about creating a script, see Microsoft Script Editor Help.

[Show All](#)

About Visual Basic scripts in a Web page

A script is a type of computer code that can be used to make your Web page dynamic. For example, a script could be used to include a "number of visitors" counter that increments each time someone visits your Web page. Or a script could be used to include a countdown to a special event: "only x more days", where x decreases by 1 every day. Creating scripts is an advanced Office feature that requires programming knowledge. If you are a developer, you can create and edit scripts directly on your Web pages in Microsoft Access. Usually scripts are run by a Web browser when a Web page is opened or by Access when in [Page view](#), typically to display information produced by the script.

Caution Be careful when copying and moving scripts. Although some scripts are self-contained and work correctly in any location on any Web page, other scripts are dependent on the structure of the Web page. When copied or moved to another Web page— or to another location on a Web page— these scripts might not run correctly or might return errors when viewed in a browser or in Page view.

[About creating and editing a Visual Basic script on a Web page](#)

[Show All](#)

Create a script on a Web page

[Microsoft Visual Basic Scripting Edition](#) (VBScript) is the default script language added to your Web page.

Note You can change the default script language.

 [How?](#)

1. Open the data access page in [Design view](#) or [Page view](#).
 2. On the **Tools** menu, point to **Macro**, and then click **Microsoft Script Editor**.
 3. In the **Properties** window, click **Document** in the list box.
 4. Change the value of the **defaultClientScript** property to either **VBScript** or **JavaScript**.
-
1. Open the Web page in [Design view](#) or [Page view](#).
 2. On the **Tools** menu, point to **Macro**, and then click **Microsoft Script Editor**.
 3. In the Microsoft Script Editor, write your script.

[Show All](#)

About Smart Tags

You can save time by using smart tags to perform [actions](#) in Microsoft Office Access 2003 that you'd usually open other programs to perform. You can attach smart tags to a file in a table or query or to controls on a form, report, or data access page.

 [How to use smart tags](#)

You add a smart tag to a field or control by setting the **Smart Tags** property. Once you have added a smart tag to a field or control, the **Smart Tag Actions**  button appears when you activate a cell in the field or the control. Click the button to see the menu of actions that are possible with the smart tag.

 [How to get more smart tags](#)

The smart tags that come with Access are just the beginning. Follow these steps to find new smart tags on the Web:

1. Create a new form in Design View.
2. Add a Text Box to the form and then click **Properties**  on the toolbar to open the control's property sheet.
3. Click the **Build** button  in the **Smart Tags** property box to open the **Smart Tags** dialog box.
4. Click **More Smart Tags**.

You may find additional smart tags created by Microsoft, by third-party companies, or by Information Technology (IT) professionals, who may design smart tags and actions for the specific products or services that you work with. For example, if you work in a sales department, you might be able to click a "product name" smart tag in your document that offers actions such as "check quantity in stock" or "check price."

 [How to get more information about developing smart tags for use with Access](#)

Visit the [Office Developer Center](#) on the Microsoft Developer Network (MSDN) Web site for more information about creating smart tags for use with Access.

Other buttons you may see in your database

At times other buttons may appear as you work in your database.

 [AutoCorrect Options button](#)

The **AutoCorrect Options**  button first appears after a word was automatically corrected and changes to a button icon when you point to it.

If you find on occasion that you don't want text to be corrected, you can undo a correction or turn AutoCorrect options off by clicking the button and making a selection.

 [Error Checking Options button](#)

The **Error Indicator**  button appears next to a control on a form or report when a common error occurs, such as two controls using the same keyboard shortcut. When you click the arrow next to **Error Indicator** , a list appears to give you options for error checking.

[Show All](#)

About the smart tags supplied by Microsoft

Microsoft provides a set of ready-to-use smart tags with Microsoft Office Access 2003. The following sections describe the tags and the actions they provide.

[About the Person Name smart tag](#)

You use the **Person Name** smart tag when you need to add to or change the contact information that you store in Microsoft Outlook.

You can take these actions with the **Person Name** smart tag:

- Send an e-mail message to a contact.
- Schedule a meeting with a contact.
- Open and edit a contact's information.
- Add the name in control to your list of contacts.

[About the Financial Symbol smart tag](#)

You use the **Financial Symbol** smart tag to look up information about a financial symbol on MSN® MoneyCentral®

You can take these actions with the **Financial Symbol** smart tag:

- Obtain a stock quote on MSN MoneyCentral.
- Obtain a report about the company on MSN MoneyCentral.
- Obtain recent news about the company on MSN MoneyCentral.

[About the Date smart tag](#)

You use the **Date** smart tag to schedule a meeting or show your Outlook calendar based on the date in the active control.

You can take these actions with the **Date** smart tag:

- Schedule a meeting.
- Display your calendar.

[Show All](#)

Add a smart tag to a control

[Add a smart tag to a control on a form, report, or data access page](#)

1. Open the form, report, or data access page in Design View.
2. Select a control and then click **Properties**  on the toolbar to open the control's property sheet.
3. In the **Smart Tags** property box, do one of the following:
 - Click the **Build** button  in the **Smart Tags** property box to open the **Smart Tags** dialog box. Click the smart tag that you want to add to the control, and then click **OK**.
 - Type the namespace Uniform Resource Identifier (URI) for the smart tag in the **Smart Tags** property field.

Note You can add a smart tag to the following control types:

[Form and report controls](#)

- Label
- Text Box
- Combo Box

[Data access page controls](#)

- Label
- Bound Span
- Text Box
- Scrolling Text
- Drop-down List
- Hyperlink

[Add a smart tag to a table field](#)

1. Open the table in Design View.
2. Select a field.
3. In the **Smart Tags** property box, do one of the following:

- Click the **Build** button  in the **Smart Tags** property box to open the **Smart Tags** dialog box. Click the smart tag that you want to add to the control, and then click **OK**.
- Type the namespace Uniform Resource Identifier (URI) for the smart tag in the **Smart Tags** property field.

[Add a smart tag to a field in a query](#)

1. Open the query in Design View.
2. Select a field in the [design grid](#) and then click **Properties**  on the toolbar to open the field's property sheet.
3. In the **Smart Tags** property box, do one of the following:
 - Click the **Build** button  in the **Smart Tags** property box to open the **Smart Tags** dialog box. Click the smart tag that you want to add to the control, and then click **OK**.
 - Type the namespace Uniform Resource Identifier (URI) for the smart tag in the **Smart Tags** property field.

[Show All](#)

Show smart tags

[Show smart tags on a form](#)

1. On the **Tools** menu, click **Options**.
2. Click the **Forms/Reports** tab.
3. Select the **Show Smart Tags on Forms** check box.
4. Click **OK**.

[Show smart tags on a datasheet](#)

1. On the **Tools** menu, click **Options**.
2. Click the **Datasheet** tab.
3. Select the **Show Smart Tags on Datasheets** check box.
4. Click **OK**.

[Show All](#)

Hide smart tags

[Hide smart tags on a form](#)

1. On the **Tools** menu, click **Options**.
2. Click the **Forms/Reports** tab.
3. Clear the **Show Smart Tags on Forms** check box.
4. Click **OK**.

[Hide smart tags on a datasheet](#)

1. On the **Tools** menu, click **Options**.
2. Click the **Datasheet** tab.
3. Clear the **Show Smart Tags on Datasheets** check box.
4. Click **OK**.

[Show All](#)

Link a table to a Windows SharePoint Services list

You can use the Link to Windows SharePoint Services Wizard in Access 2003 to create linked tables to lists or views of a list in Windows SharePoint Services.

1. In the database window, click **File** on the Access toolbar.
2. Click **Get External Data**.
3. Select **Link Tables**.
4. In the Link dialog box, for **Files of type**, select **Windows SharePoint Services**. This opens the Link to Windows SharePoint Services Wizard.
5. [On the **Select a Site** screen, specify a Windows SharePoint Services site that has the list you want to link to.](#)

You can select from the previous 10 Windows SharePoint Services sites you have connected to or type in a different site. Click **Next**.

Note You can only link tables to a site that uses Windows SharePoint Services 2.0 or later. The site must contain a list.

6. [On the **Select Lists** screen, select the lists or the views of a list you want to link to.](#)
 - If you want to link to one or more lists, select **Link to one or more lists**. Hold the SHIFT or CTRL key down to select more than one list.
 - If you want to link to one or more views of the same list, select **Link to one or more views of a list**.
 - A list contains one or more columns. A column can be defined to have lookup values that are retrieved from another list. If you want to be able to edit such a column with its lookup values, ensure that **Retrieve IDs for lookup columns** is selected. If you only want to view the column with its display value, you can clear **Retrieve IDs for lookup columns**.
7. Click **Next** and proceed to step 8, or click **Finish** to complete the linking

process.

Note If you click **Finish** at this point, linked tables will be created for the lists or the default view of the list that you have selected in step 6 above. If **Retrieve IDs for lookup columns** is selected, linked tables will also be created for any related lists that contain lookup values for columns in the selected lists or default view of the selected list.

8. [If you selected to link to one or more lists in step 6](#)

- If you have **Retrieve IDs for lookup columns** selected in step 6 above, on the **Select Related Lists** screen select the related lists that contain lookup values for columns that you would like to be able to edit. You do not need to use the shift or control key to select multiple items on this screen. Clicking an item would switch between selecting and deselecting it. Click **Next** and proceed to step 10, or click **Finish** to complete the linking process.
- If you have cleared **Retrieve IDs for lookup columns** in step 6 above, click **Next** and proceed to step 10, or click **Finish** to complete the linking process.

Note If you click **Finish** at this point, linked tables will be created for the lists that you have selected in step 6 above. If **Retrieve IDs for lookup columns** is selected, linked tables will also be created for any selected related lists.

9. [If you selected to link to one or more views of the same list in step 6](#)

1. On the **Select View** screen, select one or more views of the list to link to. Hold the shift or control key down to select more than one view. Click **Next** to proceed to the next step or click **Finish** to complete the linking process.

Note If you click **Finish** at this point, linked tables will be created for the selected views. If **Retrieve IDs for lookup columns** is selected in step 6, linked tables will also be created for any related lists that contain lookup values for columns in the selected views.

2. If you have **Retrieve IDs for lookup columns** selected in step 6, on

the **Select Related Lists** screen select the related lists that contain the lookup values for the columns that you would like to be able to edit. You do not need to use the shift or control key to select multiple items on this screen. Clicking an item would switch between selecting and deselecting it. Click **Next** and proceed to the next step, or click **Finish** to complete the linking process.

Note If you click **Finish** at this point, linked tables will be created for the selected views and the selected related lists.

3. If you have cleared **Retrieve IDs for lookup columns** in step 6 above, click **Next** and proceed to the next step, or click **Finish** to complete the linking process.

Note If you click **Finish** at this point, linked tables will be created for the selected views.

10. The **Finish** screen displays all the lists (including related lists) or views of a list that you have selected. Click **Finish** to complete the link process and have linked tables created for selected views and lists, or click **Back** to return to a previous step to make changes.

Note Some of the related lists may in turn contain columns with lookup values defined in other lists that are not displayed on this screen. You will have to go to step 1 and link to such lists separately.

[Show All](#)

How Access converts between Access and SharePoint data types

[How Access converts data types when linking to or importing the contents of a SharePoint table.](#)

The following table lists how each Windows SharePoint Services data type is converted when you link to or import a Windows SharePoint Services in Microsoft Office Access 2003.

SharePoint Data Type	Access Data Type	Default Field Property Settings	Notes
ID	AutoNumber	Field Size - Integer New Values - Increment Indexed - Yes (No Duplicates)	
Modified, Created	Date/Time (Read-only)		
Modified by, Created by			
Single line of text	Text	Field Size - 255 Default Value - Mirrors the Default Value setting in SharePoint.	The Validation Rule property will reflect the values set for the Min and Max properties in SharePoint. Access can display up to 64 kb of data in a Text Box. The displayed results are truncated if the fields contains more that 64 kb of text.
Multiple Lines of Text	Memo	Required - Mirrors Required setting in SharePoint.	

The Number of lines to display property is ignored.

Field Size - Double

Decimal Places -

Mirrors the Number of decimal places setting in SharePoint.

Default Value -

Mirrors the Default Value setting in SharePoint.

Required - Mirrors Required setting in SharePoint.

The following list illustrates how the

Format property is set according to the DefaultValue setting in SharePoint.

SharePoint Access Setting

Show as percentage

Decimal Places (when set to a number)

Decimal Places (when set to Automatic)

Default Value - Mirrors Default Value setting in SharePoint

The **Validation Rule** property will reflect the values set for the Min and Max properties in SharePoint.

Number

Number

Currency

Currency

Decimal Places -

Mirrors the Number of decimal places setting in SharePoint

Currency Format - Mirrors the Currency format setting in SharePoint

Required - Mirrors Required setting in SharePoint.

Required Mirrors the Required setting in SharePoint

The **Validation Rule** property will reflect the values set for the Min and Max properties in SharePoint.

Date/Time

Date/Time

The **Format** property is set to Short Date if the Date and Time Format is set to Date Only in SharePoint. Otherwise, the **Format** property is blank.

The following list illustrates how the **DefaultValue** property is set according to the DefaultValue setting in SharePoint.

SharePoint Access

Setting Setting

(None) blank

Today's Date =Date()

Static date Mirrors Default Value

setting in

SharePoint

		Computed value	blank
		Field Size - Long	
		Integer	
		Display Control -	
		Combo Box	
		Required - No	
		Row Source Type -	
		Table/Query Row	
		Source=SELECT ID,	
		<Lookup Field> FROM	
		<Lookup Table> Order	
		By <Lookup Field>;	
		Bound Column - 1	
		Column Count - 2	
		Column Heads - No	
		Column Widths - 0	
		List Rows - 8	
		List Width - Auto	
		Limit To List =Yes	
		Field Size - 255	
		Default Value -	
		Mirrors Default Value	
		setting in SharePoint	
		Display Control -	
		Combo Box	
		Row Source Type -	
		Item List	
		Row Source ="<choice	
		1>;"<choice 2>;..."	
		<choice N>"	
		Display Control - Text	
		Box	
		Default Value -	
		Mirrors Default Value	
		setting in SharePoint	
		Required - Mirrors	
			The Choices, Display
			choices using, and
			Allow Fill-in choices
			settings are ignored.
Lookup	Number		
Choice (single)	Text		
Choice (multiple)	Memo (Read-only in a linked table)		

		Default Value setting in SharePoint	
Grid Choice	Memo (Read-only in a linked table)	Display Control - Text Box Required - Mirrors Default Value setting in SharePoint	The Choices, Start number, and End Number settings are ignored.
Yes/No	Yes/No	Display Control - Check Box Default Value - Mirrors Default Value setting in SharePoint	
Hyperlink	Hyperlink	Required - Mirrors Default Value setting in SharePoint	The Format URL as setting is ignored.
Attachment/Picture	Hyperlink (Read-only)	Display Control - Text Box Required - Mirrors Default Value setting in SharePoint	
Computed	Can be one of the following data types: Text Number Currency Date/Time Yes/No The field is Read-only		
Rich Text	Memo	Display Control - Text Box Required - Mirrors Default Value setting in	Access can display up to 64 kb of data in a Text Box. The displayed results are truncated if the fields contains more that 64 kb of text.

SharePoint

The Number of lines to display property is ignored.

[How Access data types are converted when exporting a table to SharePoint.](#)

The following table lists how Access data types are converted when you export an Access table to SharePoint.

Access Data Type	SharePoint Data Type	Default Field Property Settings	Notes
Text	Single line of text	Column Name - Mirrors the Field Name setting in Access Description - Mirrors the Description setting in Access Required - Mirrors the Required setting in Access Maximum number of characters - Mirrors the Field Size setting in Access Default Value - Mirrors the Default Value setting in Access if it is not an expression, blank otherwise Add to Default View - Yes	The text will be truncated if it is longer than 2 ³² characters in length.
Memo	Multiple lines of text	Column Name - Mirrors the Field Name setting in Access Description - Mirrors the Description setting in Access Required - Mirrors the Required setting in Access Number of Lines to Display - 5 Add to Default View - Yes Column Name - Mirrors the Field Name setting in Access Description - Mirrors the	The text will be truncated if it is longer than 2 ³² characters in length.

Description setting in Access
Required - Mirrors the Required setting in Access
Min - Blank
Max - Blank

The following list illustrates how the **Number of decimal places** property is set according to the Decimal Places setting in Access.

Number	Number	Access Setting	SharePoint Setting
		Auto	Automatic
		0-5	0-5
		6-15	5

Default Value - Mirrors the Default Value setting in Access if it is not an expression, blank otherwise

Add to Default View - Yes

Show as percentage - Yes if the Format property is set to Percentage.

Column Name - Mirrors the Field Name setting in Access

Description - Mirrors the Description setting in Access

Required - Mirrors the Required setting in Access

Date and time format - Set to Date Only if the Format Property is set to Short Date. Otherwise, set to Date & Time.

Date/Time	Date/Time
	Calendar Type - Hijri if the Use Hijri option is checked, otherwise Gregorian.

The following list illustrates how the **Default Value** property is set according to the Default Value setting in Access.

Access Setting	SharePoint Setting
=Date()	Today's Date
Field set to a specific date	Field set to a specific date

Add to Default View - Yes

Column Name - Mirrors the Field Name setting in Access

Description - Mirrors the Description setting in Access

Required - Mirrors the Required setting in Access

Min - Blank

Max - Blank

The following list illustrates how the **Number of decimal places** property is set according to the Decimal Places setting in Access.

Currency	Currency	Access Setting	SharePoint Setting
		Auto	Automatic
		0-5	0-5
		6-15	5

Default Value - Mirrors the Default Value setting in Access if it is not an expression, blank otherwise

Add to Default View - Yes

Currency Type - Mirrors the

		Currency Format setting in Access
		Column Name - Mirrors the Field Name setting in Access
		Description - Mirrors the Description setting in Access
AutoNumber	Number	Required - Mirrors the Required setting in Access
		Min - Blank
		Max - Blank
		Number of decimal places - Automatic
		Add to Default View - Yes
		Column Name - Mirrors the Field Name setting in Access
		Description - Mirrors the Description setting in Access
AutoNumber where the Field Size property is set to Replication ID	Single line of text	Required - Mirrors the Required setting in Access
		Maximum number of characters - 38
		Default Value - Blank
		Add to Default View - Yes
		Column Name - Mirrors the Field Name setting in Access
		Description - Mirrors the Description setting in Access
Yes/No	Yes/No	Default Value - Mirrors the Default Value setting in Access if it is not an expression, blank otherwise
		Add to Default View - Yes
OLE Object	The field is not exported	
		Column Name - Mirrors the Field Name setting in Access
		Description - Mirrors the

Hyperlink

Hyperlink

Description setting in Access
Required - Mirrors the
Required setting in Access
Format URL as - Hyperlink
Add to Default View - Yes

[Show All](#)

Import from a Microsoft Windows SharePoint Services list

You can use the Import from Windows SharePoint Services Wizard in Access 2003 to import data from lists or views of a list in Windows SharePoint Services.

1. In the database window, click **File** on the Access toolbar.
2. Click **Get External Data**.
3. Select **Import**.
4. In the Import dialog box, for **Files of type**, select **Windows SharePoint Services**. This opens the Import from Windows SharePoint Services Wizard.
5. [On the **Select a Site** screen, specify a Windows SharePoint Services site that has the list you want to import.](#)

You can select from the previous 10 Windows SharePoint Services sites you have connected to or type in a different site. Click **Next**.

Note You can only import from a site that uses Windows SharePoint Services 2.0 or later. The site must contain at least one list.

6. [On the **Select Lists** screen, select the lists or the views of a list you want to import.](#)
 - If you want to import one or more lists, select **Import one of more lists**. Hold the shift or control key down to select more than one list.
 - If you want to import one or more views of the same list, select **Import one or more views of a list**.
 - A list contains one or more columns. A column can be defined to have lookup values that are retrieved from a related list. If you want to be able to edit such a column with its lookup values, ensure that **Retrieve IDs for lookup columns** is selected. If you only want to view the column with its display value, clear **Retrieve IDs for lookup columns**.

7. Click **Next** and proceed to step 8, or click **Finish** to complete the import process.

Note If you click **Finish**, the lists or the default view of the list that you have selected in step 6 above will be imported. If **Retrieve IDs for lookup columns** is selected, linked tables will be created for any related lists that contain lookup values for columns in the selected lists or default view of the selected list.

8. [If you selected to import one or more lists in step 6](#)
 - If you have **Retrieve IDs for lookup columns** selected in step 6 above, on the **Select Related Lists** screen select the related lists that contain lookup values for columns that you would like to be able to edit. You do not need to use the shift or control key to select multiple items on this screen. Clicking an item would switch between selecting and deselecting it. Then under **Link/Import Options**, choose to either link to the selected related lists, or import the selected related lists. Click **Next** and proceed to step 10, or click **Finish** to complete the import process.
 - If you have **Retrieve IDs for lookup columns** cleared in step 6 above, click **Next** and proceed to step 10, or click **Finish** to complete the import process.

Note If you click **Finish** at this point, the lists that you have selected in step 6 will be imported. If **Retrieve IDs for lookup columns** is selected, then the selected related lists will either be linked or imported, according to your choice under **Link/Import Options**.

9. [If you selected to import one or more views of the same list in step 6](#)
 1. On the **Select View** screen, select one or more views of the list to import. Hold the shift or control key down to select more than one view. Click **Next** to proceed to the next step or click **Finish** to complete the import process.

Note If you click **Finish** at this point, the selected views will be imported. If **Retrieve IDs for lookup columns** is selected in step 6, then linked tables will be created for the selected related lists.

2. If you have **Retrieve IDs for lookup columns** selected in step 6, on the **Select Related Lists** screen select the related lists that contain lookup values for columns that you would like to be able to edit. You do not need to use the shift or control key to select multiple items on this screen. Clicking an item would switch between selecting and deselecting it. Then under **Link/Import Options**, choose to either link to the selected related lists, or import the selected related lists. Click **Next** and proceed to the next step, or click **Finish** to complete the import process.

Note If you click **Finish** at this point, the selected views will be imported. The selected related lists will either be linked or imported, according to your choice under **Link/Import Options**.

3. If you do not have **Retrieve IDs for lookup columns** selected in step 6 above, click **Next** and proceed to the next step, or click **Finish** to complete the import process.

Note If you click **Finish** at this point, the selected views will be imported.

10. The **Finish** screen displays all the lists (including related lists) or views of a list that you have selected to import or link to. Click **Finish** to complete the import process and have the selected views and lists imported or linked to as you have specified, or click **Back** to return to a previous step to make changes.

Note Some of the related lists may in turn contain columns with lookup values defined in other lists that are not displayed on this screen. If you want to be able to edit with the lookup values that these lists contain, you will have to import or link to such lists separately.

Export a table or query to a Windows SharePoint Services list

You can use the Export to Windows SharePoint Services Wizard in Access 2003 to export tables or queries to lists in Windows SharePoint Services.

1. In the database window, select a table or a query.
2. On the Access toolbar, click **File**.
3. Click **Export**.
4. In the **Export Table** dialog box for the selected table or query, for **Save as type**, select **Windows SharePoint Services**. This opens the Export to Windows SharePoint Services Wizard.
5. Specify a Windows SharePoint Services site that you want to export to.

Note You can only export tables or queries to a site that uses Windows SharePoint Services 2.0 or later.

6. The name of the list defaults to the name of the table or query you are exporting. You can use the default name or type a different name.
7. Optionally, type a description for the list. This description is also the value of the description property of the table or query being exported.
8. Check **Open the list when finished** if you would like Windows SharePoint Services to open the list automatically after exporting.

Note Windows SharePoint Services opens the list in Datasheet view by default. You can change the default view of the list in Windows SharePoint Services.

9. Click **Finish**.

[Show All](#)

About applications that work with Access

Microsoft Office Access 2003 provides features for working with other programs.

- **Work with Microsoft SQL Server** You can create a Microsoft Access project that is easy to connect to a Microsoft SQL Server database, or use the Microsoft SQL Server Database Wizard to quickly create a SQL Server database and an Access project at the same time. Working with an Access project is similar to working with a Microsoft Access database—the process of creating forms, reports, data access pages, macros, and modules is the same. Once you connect to a SQL Server database, you can view, create, modify, and delete tables, views, stored procedures, and database diagrams using the Microsoft SQL Server Design Tools.
- **Create a new Access database from data in another file format** You can open a file that is in another file format— such as text, dBASE, Paradox, or spreadsheet format— in Access; Microsoft Access automatically creates an Access database and links the file for you.
- **Import or link data from Microsoft Outlook or Microsoft Exchange** You can use the Import Exchange/Outlook Wizard or the Link Exchange/Outlook Wizard to import or link data from Microsoft Exchange Server and Microsoft Outlook. For example, you might want to link to your Microsoft Outlook Contacts folder and then create form letters and mailing labels by merging the data with the Microsoft Word Mail Merge Wizard.
- **Export to, import or link data from a list in Microsoft Windows SharePoint Services** You can use various wizards in Access to share data that is stored in a list in Windows SharePoint Services:
 - The Export to Windows SharePoint Services Wizard to export an Access table to a SharePoint list.
 - The Import from Windows SharePoint Services Wizard to import data from a SharePoint list.
 - The Link to Windows SharePoint Services Wizard to link a SharePoint list to an Access table.
- **Export to other sources** You can export data to a number of formats and

applications from Access.



Application	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002, Access 2003
Microsoft Access project	9.0/2000, 10.0/Access 2002, Access 2003 III, IV, and 5; and 7 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
dBASE	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2002, Excel 2003
Microsoft Excel	3.x, 4.x, and 5.0; and 8.0 (Requires Borland Database Engine 4.x or later. Updated drivers may be available from Microsoft Technical Support)
Paradox, Paradox for Microsoft Windows	2.0
Microsoft Windows SharePoint Services	All
Microsoft Word, Rich Text Format	.wj2, .wk1 and .wk3 formats
Lotus 1-2-3	All character sets
Delimited text files	All character sets
Fixed-width text files	1.0 (if a list) 2.0, 3.x, 4.x (if a table or list)
HTML and IDC/HTX	All
Microsoft Active Server Pages	All
XML Documents	Visual FoxPro 3.0, 5.0, and 6.x
SQL tables, Microsoft Visual FoxPro, and programs and databases that support the ODBC protocol	For an up-to-date list of supported ODBC drivers, see the Microsoft Knowledge Base.

- **Import or link data to an Access database** Just as you can export to a

number of formats and applications, you can also work with the data from other sources and applications.

 [Data formats you can import or link to](#)

Data source	Version or format supported
Microsoft Access database	2.0, 7.0/95, 8.0/97, 9.0/2000, 10.0/Access 2002, Access 2003
Microsoft Access project	9.0/2000, 10.0/Access 2002, Access 2003
dBASE	III, IV, and 5; and 7 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Excel spreadsheets	3.0, 4.0, 5.0, 7.0/95, 8.0/97, 9.0/2000, and 10.0/Excel 2002, Excel 2003
Paradox, Paradox for Microsoft Windows	3.x, 4.x, and 5.0; and 8.0 (Linking requires updated ISAM drivers available from Microsoft Technical Support)
Microsoft Windows SharePoint Services	2.0
Lotus 1-2-3 spreadsheets (Linking is read-only)	.wks, .wk1, .wk3, and .wk4
Microsoft Exchange	All versions
Delimited text files	All character sets
Fixed-width text files	All character sets
HTML	1.0 (if a list) 2.0, 3.x (if a table or list)
XML Documents	Import from all versions of XML. Linking to XML data is not supported.
SQL tables, Microsoft Visual FoxPro, and data from other programs and	Visual FoxPro 2.x, 3.0, 5.0, and 6.x (import only)

databases that support the ODBC protocol

For an up-to-date list of supported ODBC drivers, see the Microsoft Knowledge Base.

[Show All](#)

About events and event procedures

Microsoft Visual Basic code is written in units called procedures. A procedure contains a series of Visual Basic statements that perform an operation or calculate a value. An event procedure is a procedure that runs in response to an event initiated by the user or program code, or triggered by the system.

[What is an event?](#)

An event is a specific action that occurs on or with a certain object. Microsoft Access can respond to a variety of events: mouse clicks, changes in data, forms opening or closing, and many others. Events are usually the result of user action.

By using an event procedure, you can add your own custom response to an event that occurs on a form, report, or control.

[What is a procedure?](#)

A procedure is a unit of [Microsoft Visual Basic](#) code. A procedure contains a series of [statements](#) and [methods](#) that perform an operation or calculate a value. For example, the following [event procedure](#) uses the **OpenForm** method to open the Orders form:

```
Private Sub OpenOrders_Click()  
  
DoCmd.OpenForm "Orders"  
End Sub
```

There are two kinds of procedures: [Sub procedures](#) and [Function procedures](#).

[Sub procedures](#)

Sub procedures perform an operation or series of operations but don't return a value. You can create your own **Sub** procedures or use the event procedure templates that Microsoft Access creates for you.

Each form and report in your database has a built-in [form module](#) or [report](#)

[module](#) that contains event procedure templates. You can add code that runs in response to events that occur on the form, the report, or the [controls](#) on the form or report. When Microsoft Access recognizes that an event has occurred on a form, report, or control, it automatically runs the event procedure named for the object and event. For example, you can write an event procedure that sets the focus to a Spouse Name text box when the user selects a Married check box.

Many wizards (for example, the Command Button Wizard) that create objects also create event procedures for the object. You can look at these event procedures to see examples of how procedures work.

[Function procedures](#)

Function procedures (often just called functions) return a value, such as the result of a calculation. Microsoft Visual Basic includes many built-in functions; for example, the **Now** function returns the current date and time. In addition to these built-in functions, you can create your own custom functions.

Because functions return values, you can use them in [expressions](#). You can use functions in expressions in many places in Microsoft Access, including in a Visual Basic statement or method, in many property settings, or in a criteria expression in a filter or query.

The following example of a **Function** procedure, `FirstOfNextMonth`, returns the date of the first day of the month following the current date:

```
Function FirstOfNextMonth()  
    FirstOfNextMonth = _  
        DateSerial(Year(Now), Month(Now) + 1, 1)  
End Function
```

This custom function consists of a single assignment statement that assigns the results of an expression (on the right side of the equal sign) to the name of the function, `FirstOfNextMonth` (on the left side of the equal sign). The function calculates a result by using the built-in Visual Basic **DateSerial**, **Year**, **Now**, and **Month** functions.

After you create this function, you can use it in an expression almost anywhere in Microsoft Access. For example, you could specify that a text box display the

first day of the month following the current date as its default value by setting the text box control's **DefaultValue** property to the following expression in the property sheet:

```
=FirstOfNextMonth()
```

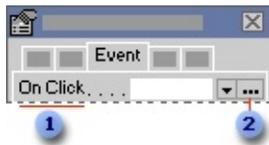
Note To use a function as a property setting, the function must be in the [form](#) or [report module](#), or in a [standard module](#). You can't use a function in a [class module](#) that isn't associated with a form or report as a form or report property setting.



[Example of using an event procedure to respond to an event](#)

When you create an event procedure for an object, Microsoft Access adds an event procedure template named for the event and the object to the form or report module. All you need to do is add code that responds in the way you want when the event occurs for the form or report.

Suppose you want a Product Details form to open when you click a Product Details command button on an Orders form. The following example shows how to do this by using an event procedure.



1 To make an event procedure run when a user clicks the command button, click the **OnClick** property of the command...

2 ... click the **Build** button, and then click **Code Builder** to add Visual Basic code to the event procedure.



1 The ProductDetails_Click event procedure uses the **OpenForm** method to open the Product Details form.

[Show All](#)

About setting the value of a control or property in response to an event

By assigning new values to [controls](#) and properties in response to [events](#) that occur on a form or report, you can make the objects you've created work together so that they're easier to use.

Here are some common situations that call for setting values with a [macro](#) or an [event procedure](#):

[Setting the value of a control on one form equal to the value of a control on another form](#)

For example, you can put an Open Orders command button on a Customers form that opens an Orders form and automatically fills in billing information from the Customers form.



1 Clicking the Open Orders button opens the Orders form with the billing information from the Customers form.

[Updating the value of a control that's dependent on the value of another control](#)

For example, you might have a PaymentDate control whose default value is the ship date (the value of the ShipDate control). You can use a macro or an event procedure to automatically reset the value of the PaymentDate control whenever the value in the ShipDate control changes.

[Setting a form or control property in response to a selection or other event](#)

You can use a macro or an event procedure to set a form or control property in order to:

- Hide or display a control based on the value of another control. For example, you might want the DateDue control on an Orders form to be hidden if the order is already paid for (if the Paid control's value is Yes) or visible if payment is still due.
- Disable or lock a control based on the value of another control. For example, you can disable a DriversLicense control if the value of an Age control is below the minimum age for a driver's license.
- Display different sets of records depending on a selection. For example, you can control which records are displayed in a Customers form, depending on which option is selected in a Customer Type option group.

[Show All](#)

Set the value of a property in response to an event

You can assign the value of a property in response to an [event](#), such as a click on a command button, by adding a SetValue action to a [macro](#) or by using an assignment [statement](#) in a [Microsoft Visual Basic](#) procedure.

[Use a macro](#)

1. To create a macro, click **Macros**  under **Objects**, and then click the **New** button on the Database window toolbar.
2. In a blank action row, click **SetValue** in the [action list](#).
3. Set the **Item** argument to the [identifier](#) for the property whose value you want to set, and then set the **Expression** argument to the value you want to assign to the property.
4. Click **Save**  to save the macro.
5. Go to the form or report, open the property sheet for the object whose event will run the macro, and set the [event property](#) to the name of the macro.

[Use Visual Basic code](#)

1. To open the event procedure for the appropriate event, open the form in [Design view](#).
2. Display the property sheet for the form and then click the **Event** tab.
3. Click the event property for the event that you want to trigger the procedure.

For example, to respond to a mouse click on a command button, open the button's OnClick event procedure.

4. Click **Build** next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Code Builder** to display the event procedure window
6. Add an assignment statement to the procedure by typing the [identifier](#) for the property whose value you want to set, an equal sign (=), and the value

you want to assign.

7. Do one of the following:

To refer to the property of a [control](#) on the current form, type the **Me** keyword followed by the **!** operator, the name of the control, the **.** (dot) operator, and the name of the property. For example, the following statement assigns the value **False** to the **Visible** property of the DateDue control on the current form:

```
Me![DateDue].Visible = False
```

To refer to the property of a control on a different form, type the control's full identifier. For example, the following identifier refers to the **Visible** property of the DateDue control on the ShipForm form:

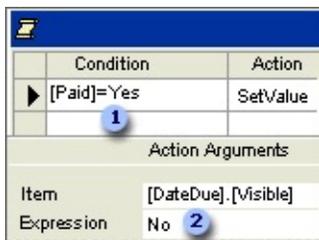
```
Forms![ShipForm]![DateDue].Visible
```

[Show All](#)

Example of setting the value of a property in response to an event

You can set the value of a property based on a value in the current record by running a [macro](#) or an [event procedure](#) in response to the form's **Current** event. For example, you might want the DateDue control on an Orders form to be hidden if the order is already paid (if the value of the Paid control is Yes or **True**).

To run a macro, set the form's **OnCurrent** event property to the name of the macro (HideDateDue) that sets the DateDue control's **Visible** property to **No** if the value of the Paid control is Yes.



- 1 If the value of the Paid control is Yes ...
- 2 ... the macro sets the DateDue control's **Visible** property to **No**.

To use an event procedure, add the following [Microsoft Visual Basic](#) code to the Form_Current event procedure:

```
If Me![Paid] = True Then
    Me![DateDue].Visible = False
Else
    Me![DateDue].Visible = True
End If
```

[Show All](#)

Order of events for database objects

A single action, such as moving from one [control](#) to another, can trigger several different [events](#), which occur in a particular sequence. Knowing when events occur and in what order they occur is important because it can affect how and when your [macros](#) or [event procedures](#) run. For example, if you have two event procedures that are to be run in a certain order, you want to make sure that the events they're associated with occur in that order.

[Order of events for controls on forms](#)

Events occur for controls on forms when you move the [focus](#) to a control, and when you change and update data in a control.

[Moving the focus to a control](#)

When you move the focus to a control on a form— for example, by opening a form that has one or more active controls or by moving to another control on the same form— the **Enter** and **GotFocus** events occur in this order:

Enter → **GotFocus**

If you are opening a form, these events occur after the events associated with opening the form (such as **Open**, **Activate**, and **Current**), as follows:

Open (form) → **Activate** (form) → **Current** (form) → **Enter** (control) → **GotFocus** (control)

When the focus leaves a control on a form— for example, when you close a form that has one or more active controls or move to another control on the same form— the **Exit** and **LostFocus** events occur in this order:

Exit → **LostFocus**

If you are closing a form, the **Exit** and **LostFocus** events occur before the events associated with closing the form (such as **Unload**, **Deactivate**, and **Close**), as follows:

Exit (control) → **LostFocus** (control) → **Unload** (form) → **Deactivate** (form)
→ **Close** (form)

[Changing and updating data in a control](#)

When you enter or change data in a control on a form and then move the focus to another control, the **BeforeUpdate** and **AfterUpdate** events occur:

BeforeUpdate → **AfterUpdate**

The **Exit** and **LostFocus** events for the changed control occur after the **BeforeUpdate** and **AfterUpdate** events:

BeforeUpdate → **AfterUpdate** → **Exit** → **LostFocus**

When you change the text in a text box or in the text box portion of a combo box, the **Change** event occurs. This event occurs whenever the contents of the control change, but before you move to a different control or record (and thus, before the **BeforeUpdate** and **AfterUpdate** events occur). The following sequence of events occurs for each key you press in a text box or in the text box portion of a combo box:

KeyDown → **KeyPress** → **Dirty** → **Change** → **KeyUp**

The **NotInList** event occurs after you enter a value in a combo box that isn't in the combo box list and then attempt to move to another control or record. It occurs after the keyboard events and the **Change** events for the combo box, but before any other control or form events. If the **LimitToList** property of the combo box is set to **Yes**, the **Error** event for the form occurs immediately after the **NotInList** event:

KeyDown → **KeyPress** → **Dirty** → **Change** → **KeyUp** → **NotInList** → **Error**

[Order of events for records on forms](#)

Events occur for records on forms when you move the [focus](#) to a record, update data in a record, delete an existing record or records, or create a new record.

[Moving the focus to records and updating data in records](#)

When you move the focus to an existing record on a form, enter or change data in the record, and then move the focus to another record, the following sequence of events occurs for the form:

Current (form) → BeforeUpdate (form) → AfterUpdate (form) → Current (form)

When you leave the record whose data has changed, but before you enter the next record, the **Exit** and **LostFocus** events occur for the control with the focus. These events occur after the **BeforeUpdate** and **AfterUpdate** events for the form, as follows:

BeforeUpdate (form) → AfterUpdate (form) → Exit (control) → LostFocus (control) → RecordExit (form) → Current (form)

As you move the focus among the controls on a form, events occur for each control. For example, the following sequences of events occur when you:

- Open a form and change data in a control:

Current (form) → Enter (control) → GotFocus (control) → BeforeUpdate (control) → AfterUpdate (control)

- Move the focus to another control:

Exit (control1) → LostFocus (control1) → Enter (control2) → GotFocus (control2)

- Move the focus to another record:

BeforeUpdate (form) → AfterUpdate (form) → Exit (control2) → LostFocus (control2) → RecordExit (form) → Current (form)

[Deleting records](#)

When you delete a record, the following events occur for the form, and Microsoft Access displays a dialog box asking you to confirm the deletion:

Delete → BeforeDelConfirm → AfterDelConfirm

If you cancel the **Delete** event, the **BeforeDelConfirm** and **AfterDelConfirm** events don't occur and the dialog box isn't displayed.

[Creating a new record](#)

When you move the focus to a new (blank) record on a form and then create a new record by typing in a control, the following sequence of events occurs:

Current (form) → **Enter** (control) → **GotFocus** (control) → **BeforeInsert** (form) → **AfterInsert** (form)

The **BeforeUpdate** and **AfterUpdate** events for the controls on the form and for the new record occur after the **BeforeInsert** event and before the **AfterInsert** event.

[Order of events for forms and subforms](#)

Events occur for forms when you open or close a form, move between forms, or work with data on a form.

[Opening and closing a form](#)

When you open a form, the following sequence of events occurs for the form:

Open → **Load** → **Resize** → **Activate** → **Current**

If there are no active controls on the form, the **GotFocus** event also occurs for the form after the **Activate** event but before the **Current** event.

When you close a form, the following sequence of events occurs for the form:

Unload → **Deactivate** → **Close**

If there are no active controls on the form, the **LostFocus** event also occurs for the form after the **Unload** event but before the **Deactivate** event.

[Moving between forms](#)

When you switch between two open forms, the **Deactivate** event occurs for the

first form, and the **Activate** event occurs for the second form:

Deactivate (form1) → **Activate** (form2)

The **Deactivate** event for a form also occurs when you switch from the form to another window in Microsoft Access. However, the **Deactivate** event doesn't occur when you switch to a dialog box, to a form whose **PopUp** property is set to **Yes**, or to a window in another application.

Note An **Open** event doesn't occur if you move to a form that is already open, even if you've moved to the form by carrying out an **OpenForm** action.

[Working with data on a form](#)

Form and control events occur as you move between records in the form and change data. For example, when you first open a form, the following sequence of events occurs:

Open (form) → **Load** (form) → **Resize** (form) → **Activate** (form) → **Current** (form) → **Enter** (control) → **GotFocus** (control)

Similarly, when you close a form, the following sequence of events occurs:

Exit (control) → **LostFocus** (control) → **Unload** (form) → **Deactivate** (form) → **Close** (form)

If you've changed data in a control, the **BeforeUpdate** and **AfterUpdate** events for both the control and the form occur before the **Exit** event for the control.

[Working with subforms](#)

When you open a form containing a subform, the subform and its records are loaded before the main form. Thus, the events for the subform and its controls (such as **Open**, **Current**, **Enter**, and **GotFocus**) occur before the events for the form. The **Activate** event doesn't occur for subforms, however, so opening a main form triggers an **Activate** event only for the main form.

Similarly, when you close a form containing a subform, the subform and its records are unloaded after the form. The **Deactivate** event doesn't occur for subforms, so closing a main form triggers a **Deactivate** event only for the main

form. The events for the controls, form, and subform occur in the following order:

1. Events for the subform's controls (such as **Exit** and **LostFocus**)
2. Events for the form's controls (including the subform control)
3. Events for the form (such as **Deactivate** and **Close**)
4. Events for the subform

Note Because the events for a subform occur after the main form is closed, certain events such as canceling the closing of the main form from an event in the subform, will not occur. You may need to move these types of validations to an event on the main form.

[Order of events for keystrokes and mouse clicks](#)

Keyboard events occur for forms and controls when you press keys or send keystrokes while the form or control has the [focus](#). Mouse events occur for forms, form sections, and controls on forms when you press the mouse buttons while the mouse pointer is on a form, section, or control, or when you move the mouse pointer over a form, section, or control.

[Keyboard events](#)

When you press and release a key while a control on a form has the focus (or use the SendKeys action or [statement](#) to send a keystroke), the following sequence of events occurs:

KeyDown → **KeyPress** → **KeyUp**

When you press and release a key or send a keystroke in the [ANSI character set](#) key, the **KeyDown**, **KeyPress**, and **KeyUp** events all occur. If you press and hold down an ANSI key, the **KeyDown** and **KeyPress** events alternate repeatedly (**KeyDown**, **KeyPress**, **KeyDown**, **KeyPress**, and so on) until you release the key; then the **KeyUp** event occurs.

If you press and release a non-ANSI key, the **KeyDown** and **KeyUp** events occur. If you press and hold down a non-ANSI key, the **KeyDown** event occurs repeatedly until you release the key; and then the **KeyUp** event occurs.

If pressing a key triggers another event for a control, the other event occurs after the **KeyPress** event but before the **KeyUp** event. For example, if a keystroke changes text in a text box, triggering a **Change** event, the following sequence of events occurs:

KeyDown → **KeyPress** → **Change** → **KeyUp**

If a keystroke causes the focus to move from one control to another control, the **KeyDown** event occurs for the first control, while the **KeyPress** and **KeyUp** events occur for the second control. For example, if you change data in a control and then press the TAB key to move to the next control, the following sequences of events occur:

- First control:

KeyDown → **BeforeUpdate** → **AfterUpdate** → **Exit** → **LostFocus**

- Second control:

Enter → **GotFocus** → **KeyPress** → **KeyUp**

 [Mouse events](#)

When you press and release a mouse button while the mouse pointer is on a control on a form, the following sequence of events occurs for the control:

MouseDown → **MouseUp** → **Click**

If a control has the focus and you click another control to move the focus to this second control, the following sequences of events occur:

- First control:

Exit → **LostFocus**

- Second control:

Enter → **GotFocus** → **MouseDown** → **MouseUp** → **Click**

If you move to another record and then click a control, the **Current** event for the

form also occurs before the **Enter** event for the control.

Double-clicking a control causes both **DbClick** and **Click** events to occur. For example, when you double-click a control other than a command button, the following sequence of events occurs for the control:

MouseDown → **MouseUp** → **Click** → **DbClick** → **MouseUp**

When you double-click a command button, these events occur, followed by a second **Click** event.

The **MouseMove** event for a form, section, or control occurs when you move the mouse pointer over the form, section, or control. This event is independent of the other mouse events.

 [Order of events for reports and report sections](#)

Events occur for reports and report sections when you open a report to print or preview it, or close a report.

 [Events for reports](#)

When you open a report to print or preview it and then later close the report or move to another window in Microsoft Access, the following sequence of events occurs for the report:

Open → **Activate** → **Close** → **Deactivate**

When you switch between two open reports, the **Deactivate** event occurs for the first report, and the **Activate** event occurs for the second report:

Deactivate (report1) → **Activate** (report2)

The **Deactivate** event for a report also occurs when you switch from the report to another window in Access. However, the **Deactivate** event doesn't occur when you switch to a dialog box, to a form whose **PopUp** property is set to **Yes**, or to a window in another application.

When you open a report that is based on a query, Access triggers the **Open** event before running the underlying query. As a result, you can set the criteria for the

report by using a macro or event procedure that responds to the **Open** event. For example, the macro or event procedure could open a custom dialog box in which you enter the report criteria.

[Events for report sections](#)

When you print or preview a report, the **Format** and **Print** events occur for the report sections after the **Open** and **Activate** events for the report and before the report's **Close** or **Deactivate** events:

Open (report) → **Activate** (report) → **Format** (report section) → **Print** (report section) → **Close** (report) → **Deactivate** (report)

In addition, the following events can occur during or after formatting but before the **Print** event:

- The **Retreat** event occurs when Access returns to a previous section during report formatting.
- The **NoData** event occurs if there are no records displayed by the report.
- The **Page** event occurs after formatting but before printing. You can use this event to customize the appearance of the printed report.

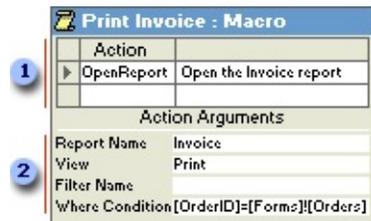
[Show All](#)

About macros and macro groups

Macros are a set of actions that you can create to help you to automate common tasks. By using groups of macros, you can perform several tasks at once.

[What is a macro?](#)

A macro is a set of one or more [actions](#) that each perform a particular operation, such as opening a form or printing a report. Macros can help you to automate common tasks. For example, you can run a macro that prints a report when a user clicks a command button.



1 When you create a macro, you enter the actions you want to carry out in this portion of the [Macro window](#).

2 You can specify arguments for an action in this portion of the window.

A macro can be one macro composed of a sequence of actions, or it can be a [macro group](#). You can also use a [conditional expression](#) to determine whether in some cases an action will be carried out when a macro runs.

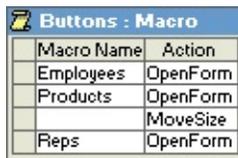
The following macro is composed of a series of actions. Microsoft Access carries out these actions each time the macro runs. To run this macro, you refer to the macro name Review Products.

Action	Comment
	Attached to the Add
Echo	Freeze screen while
OpenForm	Open the Product List
MoveSize	Position the Product L

[What is a macro group?](#)

If you have numerous macros, grouping related macros in [macro groups](#) can help you to manage your database more easily.

For example, the following macro group, named Buttons, is made up of three related macros: Employees, Products, and Reps. Each macro carries out the OpenForm action, and the Products macro also carries out the MoveSize action.



Macro Name	Action
	OpenForm
Employees	OpenForm
Products	OpenForm MoveSize
Reps	OpenForm

The name in the **Macro Name** column identifies each macro. When you run a macro in a macro group, Microsoft Access carries out the action in the action column and any actions that immediately follow with a blank **Macro Name** column.

You can run a macro in a macro group in an event or [event procedure](#) by typing the macro group name followed by a period and then the macro name. In the preceding example, to refer to the Employees macro in the Buttons macro group, you would type **Buttons.Employees**.

[Conditional actions](#)

In some cases, you may want to carry out an action or series of actions in a macro only if a particular [condition](#) is true. For example, if you're using a macro to validate data in a form, you might want to display one message in response to one set of values entered in a record and another message in response to a different set of values. In cases like these, you can use conditions to control the flow of the macro.

A condition is a logical [expression](#) that evaluates to True/False or Yes/No. The macro follows different paths depending on whether the condition is true or false.

When you run the macro, Microsoft Access evaluates the first [conditional expression](#). If the condition is true, Microsoft Access carries out the action in that row and any of the immediately following actions that are preceded by an ellipsis (...) in the **Condition** column.

Microsoft Access then runs any additional actions in the macro that have a blank **Condition** column until it reaches another expression, a macro name, or the end of the macro.

If the condition is false, Microsoft Access ignores the action and any immediately following actions that are preceded by an ellipsis in the **Condition** column and moves to the next [action row](#) that contains another condition or a blank **Condition** column.

The following macro runs the MsgBox and the StopMacro actions only when the expression in the **Condition** column is true (when there is a [Null](#) value in the SupplierID field).



Condition	Action
IsNull([SupplierID])	MsgBox
	StopMacro

[Action arguments](#)

Action arguments are additional information required by some macro actions — for example, the object affected by the action or special conditions under which the action is carried out. After you add an action to a macro, you set the arguments for the action in the lower portion of the [Macro window](#). These arguments give Microsoft Access additional information on how to carry out the action.

Tips for setting action arguments:

- In general, it's a good idea to set action arguments in the order they're listed, because choices for one argument may determine those for arguments that follow.
- If you add an action to your macro by dragging a [database object](#) from the [Database window](#), Microsoft Access automatically sets appropriate arguments for that action.
- If an action has an argument that calls for the name of a database object, you can set the argument and the corresponding object type argument automatically by dragging the object from the Database window to the argument box.
- You can use an [expression](#) preceded by an equal sign (=) to set many action

arguments.

[Show All](#)

Troubleshoot macros

[I'm using the SendKeys action in a macro to send keystrokes to a dialog box, but they're not being sent.](#)

Because the dialog box suspends the macro, you must put the SendKeys action before the action that causes the dialog box to open and set the **Wait** argument to **No** to avoid pauses in processing.

[I want to temporarily disable an action in my macro.](#)

This is similar to temporarily turning a line into a comment in [Microsoft Visual Basic for Applications](#) code by typing a single quotation mark (') at the beginning of the line.

- To cause Access to temporarily ignore an action, type **False** as a condition in the **Condition** column.

[The Where Condition argument in my macro's OpenForm or OpenReport action is producing unexpected results.](#)

The Where Condition argument of an OpenForm or OpenReport action might produce unexpected results for any of the following reasons:

- The Where Condition expression uses invalid syntax to refer to the value of a [control](#) or property.
- The Where Condition argument uses invalid syntax to combine the value of a field, control, or property with a literal string.
- The OpenForm or OpenReport action is being carried out by using the corresponding [Microsoft Visual Basic for Applications methods](#) in a procedure, and the Where Condition argument is specified in the wrong position in the list of arguments. If the Where Condition argument isn't preceded by View and Filter Name arguments, you must still type a comma as a placeholder for each of these arguments.

[The RunCode action in my macro won't run a Microsoft Visual Basic](#)

[module](#).

You use the RunCode action to run a **Function** procedure, not a [module](#). In Microsoft Access, you run procedures rather than modules. Replace the name of the module with the name of the function in the module that you want to run. To run a [Sub procedure](#), call it in the function.

To open a module, use the OpenModule action.



[When I run my macro, Access displays messages that I don't want to see.](#)

You can hide warnings and other messages while a macro is running by carrying out the SetWarnings action in the macro. You can also use the SetWarnings action to display messages again.

[Show All](#)

Create a macro or macro group

You can create a [macro](#) to perform a specific action or a [macro group](#) to perform a series of actions.

[Create a macro](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
 2. In the **Action** column, click the arrow to display the [action list](#).
 3. Click the action you want to use.
 4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
 5. Type a comment for the action. Comments are optional.
4. To add more actions to the macro, move to another [action row](#) and repeat step 3. Microsoft Access carries out the actions in the order you list them.

[Create a macro group](#)

If you want to group several related [macros](#) in one place rather than keeping track of them separately, you can organize them as a macro group.

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Click **Macro Names**  on the toolbar, if it isn't already [selected](#).
4. In the **Macro Name** column, type a name for the first macro in the macro

group.

5. Add the [actions](#) you want the macro to carry out.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
2. In the **Action** column, click the arrow to display the [action list](#).
3. Click the action you want to use.
4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
5. Type a comment for the action. Comments are optional.
6. Repeat steps 4 and 5 for any other macros you want to include in the macro group.

Notes

- When you run a macro group, Microsoft Access executes each macro, starting from the first action, until it encounters a StopMacro action, another macro group name, or no further actions.
- When you save the macro group, the name you specify is the name of the group of macros. This name is displayed in the list of macros and macro groups in the Database window. Whenever you refer to a macro in a macro group, use this syntax:

macrogroupname.macroname

For example, Buttons.Products refers to the Products macro in the Buttons macro group. In a list of macros, such as the Macro Name argument list for the RunMacro action, Microsoft Access displays the Products macro as Buttons.Products.

[Show All](#)

Modify macros

Once you have created a macro, you can modify it to make it better suit your needs.

[Add an action to a macro](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
2. In the **Action** column, click the arrow to display the [action list](#).
3. Click the action you want to use.
4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
5. Type a comment for the action. Comments are optional.

[Move an action in a macro](#)

1. In the [Macro window](#), click the [selector](#) for the [action row](#).
2. Click the selector again and drag the row to a new position.

[Show or hide macro names and conditions](#)

In one macro

Do one of the following:

- To show or hide the **Macro Name** column, click **Macro Names**  on the toolbar.
- To show or hide the **Condition** column, click **Conditions**  on the toolbar.

In all macros by default

Change whether Microsoft Access shows the **Macro Name** or **Condition** columns by default when you create macros.

1. On the **Tools** menu, click **Options**.
2. Click the **View** tab.
3. Under **Show In Macro Design**, select or clear the **Names Column** and **Conditions Column** check boxes.

[Set a macro to run under specific conditions](#)

1. In the [Macro window](#), click **Conditions**  on the toolbar.
2. In the **Condition** column, type a [conditional expression](#) in the row where you want to set the [condition](#). You can't use an SQL expression in a macro **Condition** column.

[Tip](#)

In a [Microsoft Access database](#) (.mdb), click the **Build**  button to use the Expression Builder. The Expression Builder is not available from a [Microsoft Access project](#) (.adp).

3. In the **Action** column, enter the action you want Microsoft Access to perform when the condition is true.

[Enter an expression in a macro action argument](#)

- Type an equal sign (=) followed by the [expression](#).

For example, the following expression sets the [action argument](#) to the value of the CompanyPickList [control](#):

```
=[CompanyPickList]
```

There are two instances in which you do not use an equal sign: the Expression argument of the SetValue action, and the Repeat Expression argument of the RunMacro action. If you precede an expression in these arguments with an equal sign, Microsoft Access evaluates the expression twice, which can cause

unexpected results.



In a [Microsoft Access database \(.mdb\)](#), click the **Build** button  to use the Expression Builder. The Expression Builder is not available from a [Microsoft Access project \(.adp\)](#).

[Show All](#)

Open the Macro window or the Visual Basic Editor

1. In form or report [Design view](#), do one of the following:

Double-click the [form selector](#) or [report selector](#) to open the property sheet for the form or report.

Double-click the [section selector](#) to open the property sheet for the section.

Make sure a control is selected, and then click **Properties**  on the toolbar.

2. Click the **Build** button  next to the appropriate [event property](#). For example, if you want to create a [macro](#) or [event procedure](#) that will run when you click a command button, click the **Build** button next to the command button's **OnClick** event property.
3. Click **Macro Builder** to open the [Macro window](#), or click **Code Builder** to open the [Microsoft Visual Basic Editor](#).

Note To view all the code behind a form or report, click **Code** on the **View** menu.

[Show All](#)

Convert macros to Visual Basic

Microsoft Access can automatically convert [macros](#) to [Microsoft Visual Basic event procedures](#) or [modules](#) that perform equivalent actions by using Visual Basic code. You can convert macros on a form or report, or you can convert global macros that aren't attached to a specific form or report.

[Convert macros on a form or report](#)

1. Open the form or report in [Design view](#).
2. On the **Tools** menu, point to **Macro**, and then click **Convert Form's Macros To Visual Basic** or **Convert Report's Macros To Visual Basic**.

[Convert global macros to Visual Basic](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the name of the macro you want to convert.
3. On the **File** menu, click **Save As**.
4. In the **Save As** dialog box, click **Module** in the lower text box and click **OK**.
5. In the **Convert Macro** dialog box, select the options you want, and click **Convert**.

[Show All](#)

Examples of macro conditions

You can use any [expression](#) that evaluates to True/False or Yes/No in a macro [condition](#). The macro will be executed if the condition evaluates to True (or Yes).



To cause Microsoft Access to temporarily ignore an [action](#), enter False as a condition. Temporarily ignoring an action can be helpful when you are trying to find problems in a macro.

Use this expression	To carry out the action if
[City]="Paris"	Paris is the City value in the field on the form from which the macro was run.
DCount("[OrderID]", "Orders")>35	There are more than 35 entries in the OrderID field of the Orders table.
DCount("*", "Order Details", "[OrderID]=Forms!Orders![OrderID]")>3	There are more than three entries in the Order Details table for which the OrderID field of the table matches the OrderID field on the Orders form.
[ShippedDate] Between #2-Feb-2001# And #2-Mar-2001#	The value of the ShippedDate field on the form from which the macro is run is no earlier than 2-Feb-2001 and no later than 2-Mar-2001.
Forms![Products]![UnitsInStock]<5	The value of the UnitsInStock field on the Products form is less than five.
IsNull([FirstName])	The FirstName value on the form from which the macro is run is Null (has no value). This expression is equivalent to [FirstName] Is Null .
[Country]="UK" And Forms![SalesTotals]![TotalOrds]>100	The value in the Country field on the form from which the macro is run is UK, and the value of the TotalOrds field on the SalesTotals form is greater than 100.
[Country] In ("France", "Italy", "Spain") And Len([PostalCode])<>5	The value in the Country field on the form from which the macro is run is France, Italy, or Spain, and the postal code isn't five characters long.

```
MsgBox("Confirm  
changes?",1)=1
```

You click **OK** in a dialog box in which the **MsgBox** function displays "Confirm changes?". If you click **Cancel** in the dialog box, Microsoft Access ignores the action.

About synchronizing records between two forms

You might often want to look at related records on two forms at the same time. For example, while looking at a customer record on a Customers form, you might want to open an Orders form that shows the customer's orders.

When you move between records on a form, you might also want to automatically view related records on a second form. For example, as you move between supplier records on a Suppliers form, you might want to see the products each supplier carries in a Products form.

You can synchronize records between two forms by creating a command button that opens and synchronizes a second form, or by using the **Current** event to display related records on the second form when you move between records on the first form.

You can also use the Form Wizard to create two related forms from scratch.

[Show All](#)

Synchronize two forms by using the Form Wizard

You can use the Form Wizard to create two related forms from scratch.

1. In the [Database window](#), click **Forms**  under **Objects**.
2. Click the **New** button on the Database window toolbar
3. In the **New Form** dialog box, click the **Form Wizard**.

You don't need to specify the [record source](#) for the forms here. You can do this in the wizard.

4. Click **OK**.
5. Click the name of the first table or record source that includes the data you want to base your first form on. Select the fields to be included on your form.
6. After selecting the fields from the first table or record source in the Form Wizard, repeat step 5 to select another table or record source, and pick fields that you want to include in your second form.
7. When you have selected all the required fields, click **Next**.
8. In the second screen of the wizard, click **Linked forms**, and then click **Next**.
9. Follow the instructions in the wizard. The wizard creates the two related forms; each form contains fields from one of the record sources. One of the forms also has a command button that opens the other form and synchronizes the records between the forms.

If the resulting forms don't look the way you want, you can change them in [Design view](#).

[Show All](#)

Examples of synchronizing two forms programmatically

When you move between supplier records on a Suppliers form, you might want to see the products each supplier carries in a Product List form. You can do this either with a [macro](#) or with [Microsoft Visual Basic](#) code.

[Using a macro](#)

You can accomplish this by carrying out an OpenForm [action](#) in a macro that runs in response to the Supplier form's **Current** event. You can make the Product List form display records whose value in the SupplierID field in the underlying table matches the value in the SupplierID field on the Suppliers form. Set the action's **Where Condition** argument as follows:

```
[SupplierID]=Forms![Suppliers]![SupplierID]
```

[Tip](#)

If you don't always want the Product List form to open when you move between records on the Suppliers form, include a [conditional expression](#) in the macro that checks whether or not the Product List form is open.



1 This conditional expression uses the IsLoaded function to check whether the Product List form is open. The IsLoaded function is a sample [Function procedure](#) included with the Northwind sample database, and not a built-in Visual Basic function.

After creating the macro, set the Suppliers form's **OnCurrent** event property to the name of the macro.

[Using Visual Basic code](#)

When you move between supplier records on a Suppliers form, you might want to see the products each supplier carries in a Products form. In the [Form_Current event procedure](#) of the Suppliers form, set the **FilterOn** and **Filter** properties of the Products form:

```
Private Sub Form_Current()  
  
    ' Declare and set a variable to store the WHERE  
    ' clause that describes the records you want to  
    ' display.  
    Dim strCond As String  
    strCond = "SupplierID = Forms!Suppliers!SupplierID"  
  
    ' Use the IsLoaded function from the Northwind  
    ' sample database to check whether the Products  
    ' form is open, then set the properties.  
    If IsLoaded("Products") Then  
        Forms![Products].FilterOn = True  
        Forms![Products].Filter = strCond  
    End If  
  
End Sub
```

[Show All](#)

About printing a report from a form

You can automate how a user prints a report by carrying out the OpenReport and PrintOut actions in a [macro](#) or an [event procedure](#). For example, you can make it possible for a user to print a report by clicking a button on a form, choosing a command from a custom menu, or pressing a key combination.

Use the OpenReport action if you want to restrict the records to be printed or if you want to open a report in [Print Preview](#). When you use the OpenReport action to print a report, Microsoft Access prints the report by using the default settings in the **Print** dialog box.

Use the PrintOut action to set printing options before printing your report. The PrintOut action includes [arguments](#) for every option in the **Print** dialog box.

[Show All](#)

About validating data programmatically

You can verify that data is entered correctly into a control on a form or into a field in a table by setting a property for the control, or by setting record or field [validation rules](#) in the underlying table. [Macros](#) and [event procedures](#) provide additional power and flexibility for more complex [validation](#).

Use a macro or an event procedure for data validation if:

- Your validation rule involves [conditions](#) for more than one value on the form. For example, you might want to verify that any two of three fields are filled in before saving a record.
- You want to display different error messages for different types of errors in the field. For example, if the value entered is greater than the acceptable range, you can display one message, and if it's less than the acceptable range, you can display a different message.
- You want the user to be able to override your validation rule. In this case, you can display a warning message asking the user to confirm the data.
- The validation involves references to controls on other forms or contains a function.
- You have a validation rule that you can use for more than one form, and you want the convenience of defining the rule once and then referring to it in each form.

[Using a macro](#)

In some circumstances, you might want to use validation rules that depend on complex conditions, and then take different actions depending on those conditions. For example, because different countries have different rules for their postal codes, a database that stores postal codes for many countries can't rely on a simple validation rule to ensure that a postal code is entered correctly.

You can, however, create a macro that checks which country/region is entered in the Country control and then checks that the value entered in the PostalCode

control has the right characteristics for that country/region.

First, create a macro that displays different messages depending on the value of the Country control and the length of the entry in the PostalCode control.



- 1 If the country is France and the postal code is not five characters long ...
- 2 ... the macro displays this message.

Then, open the form that contains the controls in [Design view](#), and set the **BeforeUpdate** event property of the form to the name of the macro. You use the form property rather than the control property because the validation involves two different controls on the form.

Now when you use the form to add a new supplier, Microsoft Access runs the validation macro after you finish entering the new record, but before it saves the record in the Suppliers table. If either condition in the macro is met, Access displays the appropriate message and doesn't save the record in the Suppliers table.

[Using an event procedure](#)

You can create an event procedure that checks which country/region is entered in the Country control and then checks that the value entered in the PostalCode control has the right characteristics for that country/region.

First, open the form that contains the controls in [Design view](#). Then, add the following Visual Basic code to the form's BeforeUpdate event procedure to display different messages, depending on the value of the Country control and the length of the entry in the PostalCode control.

Note You add Visual Basic code to the BeforeUpdate event procedure for the form, not the control, because the validation involves two different controls on the form.

```
Private Sub Form_BeforeUpdate(Cancel As Integer)
  Select Case Me![Country]
  Case "France"
  If Len(Me![PostalCode]) <> 5 Then
  MsgBox "PostalCode must be 5 characters."
  Cancel = True
  End If
  Case "Japan"
  If Len(Me![PostalCode]) <> 3 Then
  MsgBox "PostalCode must be 3 characters."
  Cancel = True
  End If
  End Select
End Sub
```

Now when you use the form to add a new supplier, Microsoft Access runs the validation procedure after you finish entering the new record, but before it saves the record in the Suppliers table. If either condition in the procedure is met, Access displays the appropriate message and doesn't save the record in the Suppliers table.

[Show All](#)

Validate data programmatically

You can use a [macro](#) or an [event procedure](#) to validate data being entered in a form either by creating a macro that responds to an [event](#) on the form or by creating event procedures.

[Use a macro](#)

1. To create a macro, click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
2. In the **Action** column, click the arrow to display the [action list](#).
3. Click the action you want to use.
4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
5. Type a comment for the action. Comments are optional.
4. In the [Macro window](#), make sure **Conditions**  on the toolbar is pressed in.
5. In the **Condition** column of a blank [action row](#), enter a [validation condition](#).
6. In the **Action** column, click the action in the [action list](#) that you want to carry out if the validation [condition](#) is true.
7. For any additional actions you want to carry out if the validation condition is true, enter an ellipsis (...) in the **Condition** column followed by the action in the **Action** column. For example, it's usually a good idea to carry out a `CancelEvent` action for each validation condition after any other actions have been carried out.

8. Click **Save**  to save the macro.
9. Open the form in [Design view](#).
10. For the [event](#) that you want to trigger the validation, set the [event property](#) to the name of the macro. For example, to trigger the validation when a [control](#) is updated with changed data, set the **BeforeUpdate** event property of the control to the name of the macro.

Note To validate data before a record as a whole is updated, attach the validation macro to the form's **BeforeUpdate** event rather than to the control's **BeforeUpdate** event.

[Use an event procedure](#)

1. Open the form in [Design view](#).
2. Display the property sheet for the form and then click the **Event** tab.
3. Click the event property for the event that you want to trigger the procedure.

For example, to trigger the [validation rule](#) when a [control](#) is updated with changed data, create an event procedure for the control's **BeforeUpdate** event.

4. Click **Build**  next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Code Builder** to display the event procedure window.
6. Add [Microsoft Visual Basic for Applications](#) code to the event procedure, specifying the validation condition and the operations to perform when those conditions are met.

Notes

- It's usually a good idea to cancel the event that triggers the validation after the operations have been performed. You can cancel many events by setting the event procedure's Cancel argument to True.
- To validate data before the record as a whole is updated, add code to the

BeforeUpdate event procedure for the form, rather than for the control.

[Show All](#)

Order of validation for controls and fields

[Validation rules](#) for [controls](#) and fields are evaluated in the following order:

1. [Macro](#) or [event procedure](#) validation (for example, a macro or an event procedure that responds to the **BeforeUpdate** event of the control or the form containing the control)
2. The control's **ValidationRule** property
3. The underlying table's validation rules for the field (as set in the field's **ValidationRule**, **Required**, and **AllowZeroLength** properties)
4. The underlying table's **ValidationRule** property (this property performs validation on records)

[Show All](#)

Run macros

You can run a [macro](#) directly, in a [macro group](#), from another macro or an [event procedure](#), or in response to an [event](#) that occurs on a form, a report, or a [control](#) on a form or report.

[Run a macro](#)

To run a macro directly, do one of the following:

- To run a macro from the [Macro window](#), click **Run**  on the [toolbar](#).
- To run a macro from the [Database window](#), click **Macros**  and then double-click a macro name.
- On the **Tools** menu, point to **Macro**, click **Run Macro**, and then select the macro from the **Macro Name** list.
- Run a macro from a [Microsoft Visual Basic](#) procedure by using the **RunMacro** method of the **DoCmd** object.

[Run a macro that is in a macro group](#)

To run a [macro](#) that is in a [macro group](#), do one of the following:

- Specify the macro as an [event property](#) setting on a form or report or as the Macro Name argument of the RunMacro action. Refer to the macro using this syntax:

macrogroupname.macroname

For example, this event property setting runs a macro called Categories in a macro group called Form Switchboard Buttons:

Forms Switchboard Buttons.Categories

- On the **Tools** menu, point to **Macro**, click **Run Macro**, and then select the macro from the **Macro Name** list. When macro names appear in lists, Microsoft Access includes an entry for each macro in each macro group in

the format *macrogroupname.macroname*.

- Run a macro that is in a macro group from a [Microsoft Visual Basic](#) procedure by using the **RunMacro** method of the **DoCmd** object, with the syntax shown previously to refer to the macro.

[Run a macro from another macro or from a Microsoft Visual Basic procedure](#)

Add the RunMacro action to your macro or procedure.

- To add the RunMacro action to a macro, click **RunMacro** in the [action list](#) in a blank action row, and set the **Macro Name** argument to the name of the macro you want to run.
- To add the RunMacro action to a [Visual Basic](#) procedure, add the **RunMacro** method of the **DoCmd** object to your procedure, and specify the name of the macro you want to run; for example, the following **RunMacro** method runs the macro My Macro:

```
DoCmd.RunMacro "My Macro"
```

[Run a macro or event procedure in response to an event on a form, report, or control](#)

Microsoft Access responds to many types of events on forms, reports, and controls, including mouse clicks, changes in data, and forms or reports being opened or closed.

1. Open the form or report in [Design view](#).
2. Display the property sheet for the form or report, or for a section or control on the form or report.
3. Click the **Event** tab.
4. Click the event property for the event that you want to trigger the procedure. For example, to display the event procedure for the **Change** event, click the **OnChange** property.
5. Click **Build**  next to the property box to display the **Choose Builder** dialog box.
6. Do one of following:

Create the macro

[How?](#)

1. In the **Choose Builder** dialog box, choose **Macro Builder** and click **OK**.
2. In the **Save As** dialog box, enter a name for the macro and click **OK**.
3. Add an [action](#) to the macro.

[How?](#)

1. In the [Macro window](#), click the first empty row in the **Action** column. If you want to insert an [action](#) between two [action rows](#), click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
2. In the **Action** column, click the arrow to display the [action list](#).
3. Click the action you want to use.
4. In the lower part of the window, specify arguments for the action, if any are required. For [action arguments](#) whose settings are a [database object](#) name, you can set the argument by dragging the object from the [Database window](#) to the action's **Object Name** argument box.
5. Type a comment for the action. Comments are optional.
4. To add more actions to the macro, move to another [action row](#) and repeat step 3. Microsoft Access carries out the actions in the order you list them.

Create an event procedure

[How?](#)

You can set an [event property](#) for a form, report, or [control](#) to [**Event Procedure**] to run code in response to an event. Microsoft Access creates the [event procedure](#) template for you. You can then add the code you want to run in response to the particular event.

1. Double-click **Code Builder** to display the event procedure **Sub** and **End Sub** statements in the [form module](#) or [report module](#). These statements define, or declare, the event procedure.

Microsoft Access automatically declares event procedures for each object and event in a form or report module by using the **Private** keyword to indicate that the procedure can be accessed only by other procedures in that module.

2. Add the code to the event procedure that you want to run when the event occurs. For example, to produce a sound through the computer's speaker when data in the CompanyName text box changes, add a **Beep** statement to the CompanyName_Change event procedure, as follows:

```
Private Sub CompanyName_Change()  
    Beep  
End Sub
```

The event procedure runs each time the **Change** event occurs for the object.

7. Saving the macro or procedure will set the appropriate [event property](#) to the name of the macro, or to **[Event Procedure]** if you're using an event procedure.

For example, to use a macro to display a message when you click a command button, set the command button's **OnClick** property to the name of a macro that displays the message. To use an event procedure, create a Click event procedure for the command button, and then set its **OnClick** property to **[Event Procedure]**.

[Show All](#)

Find problems in a macro by single stepping through it

By [single stepping](#) through a [macro](#), you can observe the flow of the macro and the results of each [action](#), and isolate any action that causes an error or produces unwanted results.

1. Open the macro.

[How?](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
 2. Click the name of the macro you want to open.
 3. Click the **Design** button on the Database window toolbar.
2. Click **Single Step**  on the [toolbar](#).
 3. Click **Run**  on the toolbar.
 4. Do one of the following:

To carry out the action shown in the **Macro Single Step** dialog box, click **Step**.

To stop the macro and close the dialog box, click **Halt**.

To turn off single stepping and run the remainder of the macro, click **Continue**.

[Tip](#)

You can also stop a macro while it's running and then single step through it by pressing CTRL+BREAK.

[Show All](#)

Assign an action or set of actions to a key

You can assign an [action](#) or set of actions to a specific key or key combination by creating an AutoKeys [macro group](#). When you press the key or key combination, Microsoft Access carries out the action. If you assign an action to a key combination that is already being used by Access (for example, CTRL+C is the key combination for **C**opy), the action you assign this key combination replaces the Access key assignment.

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click **New**.
3. Click **Macro Names**  on the toolbar.
4. In the **Macro Name** column, type the key or key combination to which you want to assign the action or set of actions

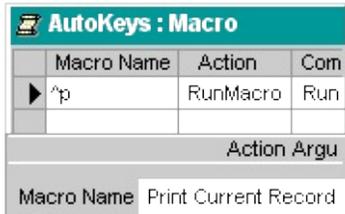
 [Syntax for AutoKeys key combinations](#)

The following table shows the key combinations you can use to make key assignments in an AutoKeys [macro group](#). These key combinations are a subset of the syntax used in the **SendKeys** statement in [Microsoft Visual Basic](#).

SendKeys syntax	Key combination
^A or ^4	CTRL+A or CTRL+4
{F1}	F1
^{F1}	CTRL+F1
+{F1}	SHIFT+F1
{INSERT}	INS
^{INSERT}	CTRL+INS
+{INSERT}	SHIFT+INS
{DELETE} or {DEL}	DEL
^{DELETE} or ^{DEL}	CTRL+DEL

+{DELETE} or +{DEL} SHIFT+DEL

5. Add the action or set of actions you want the key or key combination to carry out. For example, you could add a RunMacro action that runs the Print Current Record macro when CTRL+P is pressed.



6. Repeat steps 4 and 5 for any other key assignments you want to make.
7. Save the macro group with the name AutoKeys.

The new key assignments are in effect as soon as you save the macro group and each time you open the database.

[Show All](#)

Should I use a macro or Visual Basic?

In Microsoft Access, you can accomplish many tasks by using [macros](#) or through the user interface. In many other database programs, the same tasks require programming. Whether to use a macro or [Microsoft Visual Basic for Applications](#) often depends on what you want to do.

[When should I use a macro?](#)

Macros are an easy way to take care of simple details such as opening and closing forms and running reports. You can quickly and easily tie together the [database objects](#) you've created because there's little syntax to remember; the [arguments](#) for each action are displayed in the [Macro window](#).

In addition to the ease of use macros provide, you must use macros to:

- **Make global key assignments.**
- **Carry out an action or a series of actions when a database first opens.** However, you can use the **Startup** dialog box to cause certain things to occur when a database opens, such as opening a form.

[When should I use Visual Basic?](#)

You should use [Visual Basic](#) instead of macros if you want to:

- **Make your database easier to maintain.** Because macros are separate objects from the forms and reports that use them, a database containing many macros that respond to events on forms and reports can be difficult to maintain. In contrast, Visual Basic [event procedures](#) are built into the form's or report's definition. If you move a form or report from one database to another, the event procedures built into the form or report move with it.
- **Use built-in functions, or create your own.** Access includes many built-in functions, such as the **IPmt** function, which calculates an interest payment. You can use these functions to perform calculations without having to create complicated [expressions](#). By using Visual Basic, you can also create your own functions either to perform calculations that exceed the capability

of an expression or to replace complex expressions. In addition, you can use the functions you create in expressions to apply a common operation to more than one object.

- **Handle error messages.** When something unexpected happens while a user is working with your database, and Access displays an error message, the message can be quite mysterious to the user, especially if the user isn't familiar with Access. By using Visual Basic, you can detect the error when it occurs and either display your own message or take some action.
- **Create or manipulate objects.** In most cases, you'll find that it's easiest to create and modify an object in that object's [Design view](#). In some situations, however, you might want to manipulate the definition of an object in code. By using Visual Basic, you can manipulate all the objects in a database, as well as the database itself.
- **Perform system-level actions.** You can carry out the RunApp action in a macro to run another Microsoft Windows-based or Microsoft MS-DOS-based application from your application, but you can't use a macro to do much else outside Access. By using Visual Basic, you can check to see whether a file exists on the system, use [Automation](#) or [dynamic data exchange \(DDE\)](#) to communicate with other Windows-based applications such as Microsoft Excel, and call functions in Windows [dynamic-link libraries \(DLLs\)](#).
- **Manipulate records one at a time.** You can use Visual Basic to step through a set of records one record at a time and perform an operation on each record. In contrast, macros work with entire sets of records at one time.
- **Pass arguments to your Visual Basic procedures.** You can set arguments for macro actions in the lower part of the Macro window when you create the macro, but you can't change them when the macro is running. By using Visual Basic, however, you can pass arguments to your code at the time it is run or you can use [variables](#) for arguments— something you can't do in macros. This gives you a great deal of flexibility in how your Visual Basic [procedures](#) run.

[Show All](#)

Secure Visual Basic for Applications code in Access databases (MDB)

Note The information in this topic applies only to a Microsoft Access database (.mdb).

 [Secure Microsoft Visual Basic for Applications code with a password](#)

To prevent someone from viewing or making changes to your [Microsoft Visual Basic for Applications \(VBA\)](#) code, you can help protect the code by requiring a password.

1. Open the [Microsoft Access project \(.adp\)](#) file or [Microsoft Access database \(.mdb\)](#) file that has the VBA code you want to help protect.
2. In the Database window, point to **Macro** on the **Tools** menu, and then click **Visual Basic Editor**.

 [Tip](#)

You can also press ALT+F11.

3. In the Microsoft Visual Basic Editor, click **<Access database or Access project name> Project Properties** on the **Tools** menu.
4. On the **Protection** tab, select the **Lock project for viewing** check box. If you set a password but don't select **Lock project for viewing**, the code can be viewed and edited by anyone, but the **Project Properties** dialog box is protected.
5. In the **Password** box, type your password.

 [Guidelines for passwords](#)

Use strong passwords that combine upper- and lowercase letters, numbers, and symbols. Weak passwords don't mix these elements. Strong password: Y6dh!et5. Weak password: House27. Use a strong password that you can remember so that you don't have to write it down.

User names can range from 1 to 20 characters, and can include alphabetic characters, accented characters, numbers, spaces, and symbols, with the following exceptions:

- The characters " \ [] : | < > + = ; , . ? *
- Leading spaces
- Control characters (ASCII 10 through ASCII 31)

Note Passwords are [case-sensitive](#).

6. In the **Confirm password** box, confirm your password by typing the password again, and then click **OK**.

The password is now set. The next time you or any other user opens the database, a dialog box will be displayed that requests a password.

Important If you forget your password, it can't be recovered, and you won't be able to view or edit the VBA code.

[Create an MDE file](#)

Caution

- Be sure to save a copy of your original [Microsoft Access database](#) in a secure location. You can't modify the design of forms, reports, or modules in an Access database saved as an MDE file. If you need to change the design of these objects, you must do so in the original Access database and then resave the Access database as an MDE file.
 - You can't create an MDE file in Microsoft Access 2002 or later from a database created in an earlier version. You can only save an MDE file from a database of the same version of Access.
 - You also won't be able to convert an Access database saved as an MDE file in later versions of Access. You can run the MDE file in a later version of Access.
1. If your database enabled [user-level security](#), you must meet certain requirements before you can save it as an MDE file.

Requirements for saving a file with user-level security as an MDE file

- You must join the [workgroup information file](#) that defines the [user accounts](#) used to access the database, or that were in use when the database was created.
 - Your user account must have Open/Run and Open Exclusive [permissions](#) for the database.
 - Your user account must have Modify Design or Administer permissions for any tables in the database, or you must be the [owner](#) of any tables in the database.
 - Your user account must have Read Design permissions for all objects in the database.
2. Close the Microsoft Access database. If you are working in a multiuser environment, make sure that all other users have closed the Access database.
 3. On the **Tools** menu, click **Database Utilities**, and then click **Make MDE File**.
 4. In the **Database To Save As MDE** dialog box, specify the Access database you want to save as an MDE file, and click **Make MDE**.
 5. In the **Save MDE As** dialog box, specify a name, drive, and folder for the Access database.

[Show All](#)

Copy Visual Basic code examples from Help to the Code window

1. Select the example code you want to copy from the Help topic, and then right-click it.
2. On the shortcut menu, click **Copy**.
3. Switch to the [Microsoft Visual Basic Editor](#).
4. Paste the example into the [Code window](#) of your [module](#).

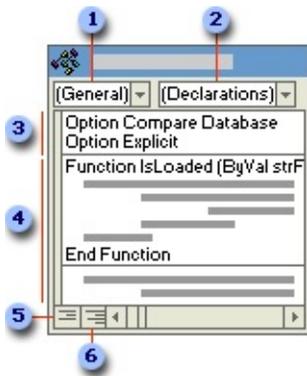
[Show All](#)

About modules

A module is essentially a collection of declarations, statements, and procedures stored together as one named unit to organize your [Microsoft Visual Basic](#) code. Microsoft Access has two types of modules: [standard modules](#) and [class modules](#).

[What is a module?](#)

A module is a collection of Visual Basic declarations and procedures that are stored together as a unit.



1 Object box

2 Procedure box

3 Declarations

4 Procedures

5 Procedure View button

6 Full Module View button

[Class modules](#)

[Form](#) and [report modules](#) are class modules that are associated with a particular

form or report. Form and report modules often contain [event procedures](#) that run in response to an event on the form or report. You can use event procedures to control the behavior of your forms and reports, and their response to user actions, such as clicking a command button.

When you create the first event procedure for a form or report, Microsoft Access automatically creates an associated form or report module.



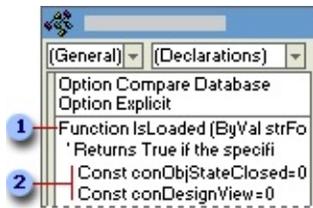
- 1 To view or add a form procedure, choose one of the form's objects from the **Object** box ...
- 2 ... and then choose an event from the **Procedure** box. Names of events that already have procedures are bold.

Procedures in your form and report modules can call procedures you have added to standard modules.

In Access 97 or later, class modules can also exist independent of a form or report, and this type of class module is listed in the [Database window](#). You can use a class module to create a definition for a custom object. In Access 95, class modules exist in association only with a form or report.

[Standard modules](#)

Standard modules contain general procedures that aren't associated with any other object, and frequently used procedures that can be run from anywhere within your database. The major difference between a standard module and a class module that is not associated with a particular object is one of scope and lifetime. The value of any variables or constants that are declared or exist in a class module without an associated object are available for use only while that code is running and only from that object.



- 1 Function name and arguments
- 2 Visual Basic declarations and code statements

Standard modules are listed in **Modules** under **Objects** in the [Database window](#).
Form, report, and standard modules are also listed in the [Object Browser](#).

[Show All](#)

About procedures

A procedure is a unit of [Microsoft Visual Basic](#) code. A procedure contains a series of [statements](#) and [methods](#) that perform an operation or calculate a value. For example, the following [event procedure](#) uses the **OpenForm** method to open the Orders form:

```
Private Sub OpenOrders_Click()  
  
DoCmd.OpenForm "Orders"  
End Sub
```

There are two kinds of procedures: [Sub procedures](#) and [Function procedures](#).

[Sub procedures](#)

Sub procedures perform an operation or series of operations but don't return a value. You can create your own **Sub** procedures or use the event procedure templates that Microsoft Access creates for you.

Each form and report in your database has a built-in [form module](#) or [report module](#) that contains event procedure templates. You can add code that runs in response to events that occur on the form, the report, or the [controls](#) on the form or report. When Microsoft Access recognizes that an event has occurred on a form, report, or control, it automatically runs the event procedure named for the object and event. For example, you can write an event procedure that sets the focus to a Spouse Name text box when the user selects a Married check box.

Many wizards (for example, the Command Button Wizard) that create objects also create event procedures for the object. You can look at these event procedures to see examples of how procedures work.

[Function procedures](#)

Function procedures (often just called functions) return a value, such as the result of a calculation. Microsoft Visual Basic includes many built-in functions; for example, the **Now** function returns the current date and time. In addition to

these built-in functions, you can create your own custom functions.

Because functions return values, you can use them in [expressions](#). You can use functions in expressions in many places in Microsoft Access, including in a Visual Basic statement or method, in many property settings, or in a criteria expression in a filter or query.

The following example of a **Function** procedure, `FirstOfNextMonth`, returns the date of the first day of the month following the current date:

```
Function FirstOfNextMonth()  
    FirstOfNextMonth = _  
        DateSerial(Year(Now), Month(Now) + 1, 1)  
End Function
```

This custom function consists of a single assignment statement that assigns the results of an expression (on the right side of the equal sign) to the name of the function, `FirstOfNextMonth` (on the left side of the equal sign). The function calculates a result by using the built-in Visual Basic **DateSerial**, **Year**, **Now**, and **Month** functions.

After you create this function, you can use it in an expression almost anywhere in Microsoft Access. For example, you could specify that a text box display the first day of the month following the current date as its default value by setting the text box control's **DefaultValue** property to the following expression in the property sheet:

```
=FirstOfNextMonth()
```

Note To use a function as a property setting, the function must be in the [form](#) or [report module](#), or in a [standard module](#). You can't use a function in a [class module](#) that isn't associated with a form or report as a form or report property setting.

[Show All](#)

About working with data and database objects by using Visual Basic code

In addition to working with data and objects such as forms and reports in the Microsoft Access interface, you can use [Microsoft Visual Basic](#) to write [procedures](#) that dynamically create, delete, and modify data and objects. For example, you can write a procedure that places a caption on a form or that changes the color of a field on a form when a user performs a particular action.

You can refer to data and objects directly in your code, or you can declare object [variables](#) to represent them. After an object variable is declared and assigned, you can use it just as you would the name of the object it represents, and you can change its value, just as you can change the value of any variable.

Note Before running the following example, make sure that you have set a reference to Microsoft ADO Ext. 2.1 (or later) for DDL and Security in the [Microsoft Visual Basic Editor](#).

 [How?](#)

1. Open the Module window.
2. On the **Tools** menu, click **References**.
3. In the **Available References** box, select **Microsoft ADO Ext. 2.1 (or later) for DDL and Security**, and then click **OK**.

The following [Sub procedure](#) creates a table, Old Invoices, with a single field, OrderID. It provides an example of declaring objects, assigning a value, and referring to those objects in code:

```
Public Sub ADOXCreateTable()
```

```
' The ADOX Catalog object corresponds to the DAO Database
```

```

' object
Dim cat As ADOX.Catalog
Dim tbl As ADOX.Table

' Open the catalog
cat.ActiveConnection = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=<full path to your database>\<your database name>"

' Create a new Table object.
With tbl
    .Name = "Old Invoices"
    ' Create a field and append it to the new Table
    ' object. This must be done before appending the
    ' Table object to the Tables collection of the
    ' Catalog.
    .Columns.Append "OrderID", adVarChar
End With

' Add the new table to the database.
cat.Tables.Append tbl

'Clean up the references
Set cat = Nothing
Set tbl = Nothing

End Sub

```

Microsoft [ActiveX Data Objects \(ADO\)](#) provides the objects, such as tables, queries, [relationships](#), and [indexes](#), that handle data-management tasks in a [Microsoft Access database](#). These objects are called data access objects. You can share Visual Basic code that uses data access objects with other applications that use Microsoft ADO, such as Microsoft Excel.

You can also use [Jet and Replication Objects \(JRO\)](#) to access and manipulate data in a database server through any [OLE DB provider](#).

In addition, Microsoft Access defines a variety of objects that you use for working with data, such as forms, reports, and [controls](#).

[Show All](#)

Change the display of your Visual Basic code

 [Use the Immediate window of the Microsoft Visual Basic Editor](#)

Use the [Immediate window](#) of the [Microsoft Visual Basic Editor](#) to check the results of a line of [Microsoft Visual Basic](#) code. You can use the Immediate window to check the value of a control, field, or [property](#); to display the result of an [expression](#); or to assign a new value to a [variable](#), [field](#), [column](#), or property. The Immediate window is a kind of scratchpad window in which [statements](#), [methods](#), and [Sub procedures](#) are evaluated immediately.

1. Do one of the following:

To display the Immediate window at any time in Microsoft Access, press CTRL+G.

To use the Immediate window at a certain point during execution of the code, suspend execution at that point and then click **Immediate Window**  on the **Debug** toolbar.

2. Do one of the following:

Type a statement, method, **Function**, or **Sub** procedure call in the window, and then press ENTER.

View the results of an expression in the Immediate window by entering the **Print** method of the **Debug** object, followed by the expression. You can use a question mark (?) as shorthand for the **Print** method.

For example, Microsoft Visual Basic displays the value returned by the FirstOfNextMonth custom function if you type either of the following lines in the Immediate window:

```
Debug.Print FirstOfNextMonth()
```

```
? FirstOfNextMonth()
```

[Change how Visual Basic code is displayed in the Code window](#)

You can change the color, font, font size, and tab width used when you type [Visual Basic](#) code in the [Code window](#). You can also specify whether the entire [module](#) is displayed (default) or each [procedure](#) is displayed individually in the Code window, and set various additional options for working in the Code window.

1. On the **Tools** menu, click **Options**.
2. Set the appropriate options.

[Display a list of relevant objects, properties, and methods in the Visual Basic Editor](#)

When **Auto List Members** is selected, Microsoft Access assists you in completing [statements](#) while you're writing [Visual Basic](#) code.

1. On the **Tools** menu, click **Options**.
2. Click the **Editor** tab.
3. Under **Code Settings**, make sure **Auto List Members** is selected.

Note To make the list disappear, press ESC.

[Display syntax information in the Visual Basic Editor](#)

You can set the [Visual Basic Editor](#) to display syntax information in the [Code window](#). When you type a [method](#) or the name of an existing [procedure](#) followed by a space or opening parenthesis, a tip automatically appears giving syntax information about the method or procedure, such as the [arguments](#) for you to specify.

1. On the **Tools** menu, click **Options**.
2. Click the **Editor** tab.
3. Under **Code Settings**, make sure **Auto Quick Info** is selected.

[Display the value of a variable or expression in the Visual Basic Editor when code is suspended](#)

You can check the value of a [variable](#) or an [expression](#) while code is suspended. Rest the mouse pointer on a variable or expression to display its current value.

1. On the **Tools** menu, click **Options**.
2. Click the **Editor** tab.
3. Under **Code Settings**, make sure **Auto Data Tips** is selected.

[Display the value of variables in the Locals window of the Visual Basic Editor](#)

In the [Locals window](#), Microsoft Access automatically displays the name, current value, and type of all the variables and objects in the current [procedure](#). The values in the Locals window are updated each time you suspend code execution. You can change the value of a variable in the Locals window by selecting the existing value and entering the new value.

1. Suspend execution of [Visual Basic](#) code.

[How?](#)

When you suspend execution of [Microsoft Visual Basic](#) code, the code is still running but is paused between running [statements](#). While your code is suspended, you can do [debugging](#) tasks such as checking current [variable](#) values and running each line of code one at a time. To make Microsoft Visual Basic pause your code, you can set a [breakpoint](#).

1. In the [Visual Basic Editor](#), move the insertion point to a line of code that isn't already a breakpoint or declaration.
2. Click **Toggle Breakpoint**  on the **Debug** toolbar.
3. Do one of the following:

To clear a breakpoint, move the insertion point to the line of code on which the breakpoint has been set, and then click **Toggle Breakpoint**  on the **Debug** toolbar.

To resume running the code, click **Run Sub/UserForm** on the **Run** menu.

Note You can also suspend code by adding a **Stop** statement to a procedure

or by pressing CTRL+BREAK while the code is running.

2. Click **Locals Window** on the **View** menu.

[Show All](#)

Create a Visual Basic procedure

By extending your database to include [Microsoft Visual Basic procedures](#), you can customize the way the tables, forms, reports, and queries in your database work together. There are several types of procedures. You can create an [event procedure](#) by adding code to an event on a form or report. You also can create your own [Function procedures](#) or [Sub procedures](#) in [standard modules](#), or in [class modules](#) (which include [form](#) and [report modules](#)).

[Create a custom function](#)

1. To open a [module](#), do one of the following:

To open a new [standard module](#), in the Database window, click **Modules**  under **Objects**, and then click the **New** button on the Database window toolbar.

To open an existing standard module, click **Modules**  under **Objects**, select the module you want to open, and then click **Design**.

To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the toolbar.

To open a new [class module](#) that isn't associated with a form or report, in the Database window click **Class Module** on the **Insert** menu.

To open an existing class module, in the Database window, click **Modules**  under **Objects**, select the module you want to open, and then click the **Design** button on the Database window toolbar.

2. Declare the function by typing the **Function** statement.
3. Type a function name immediately followed by any function [arguments](#) in parentheses. For example, the following declaration for the **IsLoaded** function specifies *strFormName* as an argument:

```
Function IsLoaded (strFormName As String) As Boolean
```

4. Add the [Microsoft Visual Basic](#) code that performs the operation or calculation that you want the function to perform.

[Create a custom Sub procedure](#)

1. To open a [module](#), do one of the following:

To open a new [standard module](#), in the Database window, click **Modules**  under **Objects**, and then click **New** on the Database window toolbar.

To open an existing standard module, click **Modules**  under **Objects**, select the module you want to open, and then click **Design**.

To open a [form module](#) or [report module](#), open the form or report in [Design view](#), and then click **Code**  on the toolbar.

To open a new [class module](#) that isn't associated with a form or report, in the Database window click **Class Module** on the **Insert** menu.

To open an existing class module, in the Database window, click **Modules** , select the module you want to open, and then click **Design** on the Database window toolbar.

2. Declare the procedure by typing the **Sub** statement.
3. Type a procedure name, immediately followed by any [arguments](#) in parentheses. For example, the following declaration for the ShowEvent [Sub procedure](#) specifies *EventName* as an argument:

```
Sub ShowEvent(EventName As String)
```

4. Add the [Microsoft Visual Basic](#) code that performs the operation that you want the procedure to perform.

[Create an event procedure](#)

You can set an [event property](#) for a form, report, or [control](#) to **[Event Procedure]** to run code in response to an event. Microsoft Access creates the [event procedure](#) template for you. You can then add the code you want to run in response to the particular event.

1. Open a form or report in [Design view](#).
2. Display the property sheet for the form or report, or for a section or control on the form or report.
3. Click the **Event** tab.
4. Click the event property for the event that you want to trigger the procedure. For example, to display the event procedure for the **Change** event, click the **OnChange** property.
5. Click **Build** next to the property box to display the **Choose Builder** dialog box.
6. Double-click **Code Builder** to display the event procedure **Sub** and **End Sub** statements in the [form module](#) or [report module](#). These statements define, or declare, the event procedure.

Microsoft Access automatically declares event procedures for each object and event in a form or report module by using the **Private** keyword to indicate that the procedure can be accessed only by other procedures in that module.

7. Add the code to the event procedure that you want to run when the event occurs. For example, to produce a sound through the computer's speaker when data in the CompanyName text box changes, add a **Beep** statement to the CompanyName_Change event procedure, as follows:

```
Private Sub CompanyName_Change()  
    Beep  
End Sub
```

The event procedure runs each time the **Change** event occurs for the object.

[Create a class module that is not associated with a form or report](#)

1. In the [Database window](#) or in the [Microsoft Visual Basic Editor](#), click **Class Module** on the **Insert** menu.

An empty class module appears in the Visual Basic Editor.

2. Add any declarations and procedures that you want to the [module](#).

3. To save the module, click **Save** on the [toolbar](#), and specify a name for the class module in the **Save As** dialog box.

[Show All](#)

Example of using Visual Basic to change a query's SQL statement

You can use the **SQL** property of a **QueryDef** object to change the underlying SQL statement of a query. For example, the following block of [Microsoft Visual Basic](#) code changes the underlying SQL statement of the query Employee List so that the query selects all records in the Employees table:

```
Dim dbsCurrent As Database
Dim qryTest As QueryDef

Set dbsCurrent = CurrentDb
Set qryTest = dbsCurrent.QueryDefs("Employee List")
qryTest.SQL = "SELECT * FROM Employees;"
```

[Show All](#)

Get information about the objects used in your Visual Basic code

You can use the [Object Browser](#) to view and navigate among the objects available in Microsoft Access and other applications that support [Microsoft Visual Basic](#), as well as the methods and properties of each object. When you locate a method or property in the Object Browser, you can paste it into the active [module](#).

[Locate a method or property](#)

1. Open any module in the [Microsoft Visual Basic Editor](#).
2. Click **Object Browser**  on the toolbar.
3. In the **Project/Library** box (the upper-left box), click **Access** or another library.
4. In the **Classes** box, click the object you're interested in.
5. In the **Members Of** box, click the method or property you want.

[View Help on a class, method, event, or property](#)

- Select the class, method, event, or property and click **Help**  in the Object Browser.

[View objects in the Office 2003 Object Library](#)

To view objects provided by the Microsoft Office Object Library, such as command bar objects, you must set a reference to this library.

1. In the [Microsoft Visual Basic Editor](#), click **References** on the **Tools** menu.
2. Select the **Microsoft Office 11.0 Object Library** check box.

[Paste a method or property syntax into a module](#)

1. Select the method or property in the Object Browser.
2. In the bottom section of the Object Browser, select the syntax you want to

- paste.
3. Click **Copy To Clipboard**  in the Object Browser.
 4. Click in the [Code window](#) of the module where you want to paste the syntax.
 5. Click **Paste**  on the toolbar.

Microsoft Access pastes the method or property syntax into the module.

Note After pasting syntax for [Sub procedures](#), you need to remove any parentheses surrounding placeholders in the syntax.

 [Display relevant objects, properties, and methods in the Microsoft Visual Basic Editor](#)

You can get assistance while you're writing Visual Basic code. When you type the name of an object, followed by a space or opening parenthesis, Microsoft Access displays a list of relevant objects, [properties](#), and methods that could follow the object name.

1. Open any module in the [Microsoft Visual Basic Editor](#).
2. On the **Tools** menu, select **Options**, click the **Editor** tab, and then select the **Auto List Members** check box. There are several keys you can use to enter an item from the list into the current line of code.

Key	Effect
TAB	Enters the item
SPACEBAR	Enters the item followed by a space
ENTER	Enters the item and moves the cursor to the next line

You can make the list disappear by pressing ESC.

[Show All](#)

Guidelines for naming Visual Basic procedures, variables, and constants

Names of [procedures](#), [variables](#), and constants in your [Microsoft Visual Basic for Applications](#) code:

- Can be up to 255 characters long.
- Must begin with a letter.
- Can include letters, numbers, or underscore characters (_).
- Can't include punctuation characters or spaces.
- Can't be a Visual Basic keyword. A keyword is a word that Visual Basic uses as part of its language. This includes predefined statements (such as **If** and **Loop**), functions (such as **Len** and **Abs**), and operators (such as **Or** and **Mod**).

[Show All](#)

Require variable declarations for Visual Basic code

[Microsoft Visual Basic](#) doesn't automatically require that you explicitly declare a [variable](#) before using it in a [procedure](#). If you use a variable that hasn't been explicitly declared, Visual Basic implicitly declares it as a variable with the **VARIANT** data type. Although implicit declarations are convenient, they can lead to subtle errors in your code.

You can require that variables be declared before they are used in a procedure. Microsoft Access automatically includes an **Option Explicit** statement in the [Declarations section](#) of all new [modules](#) in the database, including the [form](#) and [report modules](#) associated with new forms or reports.

1. On the **Tools** menu, click **Options**.
2. Click the **Editor** tab.
3. Under **Code Settings**, select the **Require Variable Declaration** check box.

[Show All](#)

Run Visual Basic code

You run [Microsoft Visual Basic](#) code in Microsoft Access by running a **Sub procedure** or **Function procedure**. Procedures contain a series of [statements](#) and [methods](#) that perform an operation or calculate a value.

Procedures are stored in units called [modules](#). However, you don't run a module; rather, you call the procedures from [expressions](#), other procedures, or [macros](#). [Event procedures](#) in [form](#) or [report modules](#) also run in response to user actions.

To run Visual Basic code in Microsoft Access, do one of the following:

- Create an event procedure. When the user performs an action that causes that event, the event procedure runs.

For example, you can add code to the Click event procedure of a command button that opens a form when a user clicks the button.

- Use a function in an expression or in the [Immediate window](#) of the [Microsoft Visual Basic Editor](#).

For example, you can use a function in an expression that defines a [calculated field](#) in a form, report, or query. You can use expressions as property settings in queries and filters, in macros and actions, in Visual Basic statements and methods, or in SQL statements.

- Call a **Sub** procedure in another procedure or in the Immediate window of the Visual Basic Editor.

If you have code that you run frequently, you can put it in a **Sub** procedure. Rather than repeating the Visual Basic code that performs the operation in each procedure, you write it once in the common procedure and then call the common procedure each time you want to perform the operation.

- Run a procedure that does not take [arguments](#). In the [Code window](#) of the Visual Basic Editor, place the cursor in the procedure that you want to run

and then click **Run Sub/UserForm** on the **Run** menu.

- Carry out the RunCode action in a macro.

You can use the RunCode action to run a built-in Visual Basic function or a function you've created. To run a **Sub** procedure or an event procedure, create a function that calls the **Sub** procedure or event procedure, and then use the RunCode action to run the function.

[Show All](#)

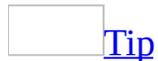
Use arguments in Visual Basic

- Specify the [arguments](#) immediately after the name of the [procedure](#), [statement](#), or [method](#). For a **Function procedure** or a method that returns a value, enclose the arguments in parentheses. If you specify more than one argument, separate the arguments with a comma.

For example, the following **Close** method of the **DoCmd** object closes the Add Products form. It accepts two arguments, `acForm` and `"Add Products"`:

```
DoCmd.Close acForm, "Add Products"
```

Because the **Close** method doesn't return a value, the arguments aren't enclosed in parentheses.



You can use the [Object Browser](#) to paste method or property syntax, including arguments, into your [Microsoft Visual Basic](#) code. The Object Browser pastes named arguments that you can specify in any order. You can use these named arguments, or you can replace the named arguments with standard arguments.

[Show All](#)

About errors in Visual Basic code

Debugging is a process you use to find and resolve errors, or bugs, in your [Microsoft Visual Basic](#) code. There are three types of errors you may encounter as your code runs:

- [Compile-time](#) errors occur as a result of incorrectly constructed code. You may have forgotten to balance pairs of statements (such as **If** and **End If** or **For** and **Next**), or you may have a programming mistake that violates the rules of Microsoft Visual Basic (such as a misspelled word, missing separator, or type mismatch error).

Compile-time errors also include [syntax errors](#), which are errors in grammar or punctuation. These include mismatched parentheses or an incorrect number of [arguments](#) passed to a function.

- [Run-time errors](#) occur after the application starts to run. Examples of run-time errors include attempting an illegal operation, such as dividing by zero or writing to a file that doesn't exist.
- Logic errors occur when the application doesn't perform as intended and produces incorrect results.

To help you isolate all three types of errors and to monitor how your code runs, the [Microsoft Visual Basic Editor](#) provides debugging tools that let you step through your code one line at a time, examine or monitor the values of expressions and [variables](#), and trace procedure calls.

[Show All](#)

Debug your Visual Basic code

[Add a breakpoint to suspend execution of Microsoft Visual Basic code](#)

When you suspend execution of [Microsoft Visual Basic](#) code, the code is still running but is paused between running [statements](#). While your code is suspended, you can do [debugging](#) tasks such as checking current [variable](#) values and running each line of code one at a time. To make Microsoft Visual Basic pause your code, you can set a [breakpoint](#).

1. In the [Visual Basic Editor](#), move the insertion point to a line of code that isn't already a breakpoint or declaration.
2. Click **Toggle Breakpoint**  on the **Debug** toolbar.
3. Do one of the following:

To clear a breakpoint, move the insertion point to the line of code on which the breakpoint has been set, and then click **Toggle Breakpoint**  on the **Debug** toolbar.

To resume running the code, click **Run Sub/UserForm** on the **Run** menu.

Note You can also suspend code by adding a **Stop** statement to a procedure or by pressing CTRL+BREAK while the code is running.

[Step through Visual Basic code](#)

Stepping through your [Visual Basic](#) code can help you identify where an error is occurring. You can see whether each line of code produces the results you expect.

1. Suspend execution of the code.

[How?](#)

When you suspend execution of [Microsoft Visual Basic](#) code, the code is still running but is paused between running [statements](#). While your code is

suspended, you can do [debugging](#) tasks such as checking current [variable](#) values and running each line of code one at a time. To make Microsoft Visual Basic pause your code, you can set a [breakpoint](#).

1. In the [Visual Basic Editor](#), move the insertion point to a line of code that isn't already a breakpoint or declaration.
2. Click **Toggle Breakpoint**  on the **Debug** toolbar.
3. Do one of the following:

To clear a breakpoint, move the insertion point to the line of code on which the breakpoint has been set, and then click **Toggle Breakpoint**  on the **Debug** toolbar.

To resume running the code, click **Run Sub/UserForm** on the **Run** menu.

Note You can also suspend code by adding a **Stop** statement to a procedure or by pressing CTRL+BREAK while the code is running.

Microsoft Access displays the line of code where execution is suspended.

2. Do one of the following:

To step through each line of code and into the code in a procedure called by another procedure, click **Step Into**  on the **Debug** toolbar.

To step through each line of code but run as a unit any procedure that is called, click **Step Over**  on the **Debug** toolbar.

To run the code that precedes the current line of code, and then break so you can step through each line of code, click **Run To Cursor** on the **Debug** menu.

To run the rest of the current procedure, and then return to the next line of code in the preceding procedure in the [call tree](#), click **Step Out**  on the toolbar.

Note You can use **Step Into**  without suspending code execution. In the [Visual Basic Editor Code window](#), place your cursor in the procedure you want

to step through, and click **Step Into**  on the toolbar.

You can alternate between these types of stepping. The type of stepping you do depends on which portions of code you want to analyze.

[Perform a quick watch while debugging Visual Basic code](#)

1. Suspend execution of your [Visual Basic](#) code.

[How?](#)

When you suspend execution of [Microsoft Visual Basic](#) code, the code is still running but is paused between running [statements](#). While your code is suspended, you can do [debugging](#) tasks such as checking current [variable](#) values and running each line of code one at a time. To make Microsoft Visual Basic pause your code, you can set a [breakpoint](#).

1. In the [Visual Basic Editor](#), move the insertion point to a line of code that isn't already a breakpoint or declaration.
2. Click **Toggle Breakpoint**  on the **Debug** toolbar.
3. Do one of the following:

To clear a breakpoint, move the insertion point to the line of code on which the breakpoint has been set, and then click **Toggle Breakpoint**  on the **Debug** toolbar.

To resume running the code, click **Run Sub/UserForm** on the **Run** menu.

Note You can also suspend code by adding a **Stop** statement to a procedure or by pressing CTRL+BREAK while the code is running.

Microsoft Access displays the line of code where execution is suspended.

2. Select the [expression](#) whose value you want to see.
3. Click **Quick Watch**  on the **Debug** toolbar.

Microsoft Access displays the **Quick Watch** dialog box, where you can see the expression and the current value of the expression. Click **Add** in the dialog box

to add the expression to the list of watch expressions in the [Watch window](#) of the [Microsoft Visual Basic Editor](#).

[Trace Visual Basic procedure calls while debugging Visual Basic code](#)

When you suspend execution of [Visual Basic](#) code while [debugging](#) the code, you can use the **Calls** dialog box to view a list of procedures that have been started but not completed.

1. Suspend execution of your Visual Basic code.

[How?](#)

When you suspend execution of [Microsoft Visual Basic](#) code, the code is still running but is paused between running [statements](#). While your code is suspended, you can do [debugging](#) tasks such as checking current [variable](#) values and running each line of code one at a time. To make Microsoft Visual Basic pause your code, you can set a [breakpoint](#).

1. In the [Visual Basic Editor](#), move the insertion point to a line of code that isn't already a breakpoint or declaration.
2. Click **Toggle Breakpoint**  on the **Debug** toolbar.
3. Do one of the following:

To clear a breakpoint, move the insertion point to the line of code on which the breakpoint has been set, and then click **Toggle Breakpoint**  on the **Debug** toolbar.

To resume running the code, click **Run Sub/UserForm** on the **Run** menu.

Note You can also suspend code by adding a **Stop** statement to a procedure or by pressing CTRL+BREAK while the code is running.

Microsoft Access displays the line of code where execution is suspended.

2. Click **Call Stack**  on the **Debug** toolbar.

Microsoft Access displays the most recently called procedure at the top of the

list, followed by the next most recent procedure call, and so on. To display the statement that calls the next procedure in the list, click **Show**.

[Add, modify, or delete a watch expression in the Watch window of the Microsoft Visual Basic Editor](#)

[Add a watch expression](#)

1. Open a procedure in the [Code window](#).
2. On the **Debug** menu, click **Add Watch**.
3. If you already selected an [expression](#) in the Code window, it is automatically displayed in the dialog box. If no expression is displayed, type the expression you want to evaluate. The expression can be a [variable](#), a [property](#), a function call, or any other valid expression.

Instead of typing the expression, you can also select an expression in the Code window and drag it to the [Watch window](#).

4. To select the range for which the expression will be evaluated, select a [module](#) and procedure context under **Context**.

Note Select the narrowest scope that fits your needs. Selecting all procedures or all modules could slow down code execution.

5. To define how the system responds to the watch expression, select an option under **Watch Type**.

To display the value of the watch expression, click **Watch Expression**.

To pause execution if the expression evaluates to **True**, click **Break When Value Is True**.

To pause execution if the value of the expression changes, click **Break When Value Changes**.

When you run the code, the Watch window shows the value of the expression you set.

[Modify a watch expression](#)

1. Select the [expression](#) in the [Watch window](#).
2. Click **Edit Watch** on the **Debug** menu.

[Delete a watch expression](#)

1. On the **Debug** menu, click **Edit Watch**.
2. Click **Delete** in the **Edit Watch** dialog box.

[Disable syntax checking](#)

[Microsoft Visual Basic](#) automatically checks the code you type in the [Code window](#) of the [Microsoft Visual Basic Editor](#) for syntax errors as you type it. You can disable this feature.

1. On the **Tools** menu, click **Options**.
2. Click the **Editor** tab.
3. Under **Code Settings**, clear the **Auto Syntax Check** check box.

[Ignore error handling](#)

When you add an **On Error** statement to a Visual Basic procedure, Visual Basic automatically branches to an error-handling routine when an error occurs. You may want to ignore a procedure's error-handling code in some circumstances — for example, while you're debugging the procedure.

1. On the **Tools** menu, click **Options**.
2. Click the **General** tab.
3. Under **Error Trapping**, select **Break On All Errors**.

When you check this option, you'll automatically enter Break mode when an error occurs, regardless of error handling.

[Show All](#)

Improve Visual Basic code performance

Often, the best way to improve the speed at which your [Microsoft Visual Basic](#) code runs is to choose a more efficient approach. However, certain techniques can help make your code run faster.

- Make sure the code in your database is compiled. Renaming a database decompiles the code in it so recompile your code if you rename the database.
- Always explicitly declare [variables](#). You can require that variables in your database be explicitly declared before they are used in a [procedure](#).
- Use the most specific type possible when you declare variables. For example, declare a variable that will be used to represent a form as type **Form** rather than as type **Object** or **Variant**.
- Use variables to refer to [properties](#), controls, and data access objects. If you refer more than once to the value of a property or control on a form, or to a data access object or its property, create object variables and refer to the variables rather than using full identifiers.
- Use the **Me** keyword for form references within an [event procedure](#).
- Use the **IIf** function sparingly. Avoid using the **IIf** function if either of the return expressions takes a long time to evaluate.
- Use string functions when appropriate. Use the **\$** functions (for example, the **Str\$** function) when working with strings and the non-**\$** functions when working with **Variant** or other data types.
- Use the **Integer** or **Long** data type for math when possible.
- Use dynamic arrays and the **Erase** or **ReDim** statement to reclaim memory.
- Use [constants](#) whenever possible.
- Use bookmarks for fast relocation. Use bookmarks instead of the **FindNext** method or some other means to return to a particular record.
- Use the **FindRecord** and **FindNext** methods only on indexed fields. These methods are much more efficient when used on a field that is indexed.
- Organize your modules. Visual Basic loads modules on demand—that is, it loads a module into memory only when your code calls one of the procedures in that module. Placing related procedures in the same module

causes Visual Basic to load additional modules only as needed.

- Eliminate unused procedures and unused variables. As you develop and modify your applications, you may leave behind procedures that aren't called from anywhere in your code or declared variables that are no longer used. Review your code frequently to find and remove unused procedures and variables.

[Show All](#)

Troubleshoot Visual Basic code

[When I run my code, Microsoft Access displays messages that I don't want to see.](#)

You can temporarily stop the display of warnings and other messages while [Microsoft Visual Basic](#) code is running by using the **SetWarnings** method to carry out the SetWarnings action. You can also use the **SetWarnings** method to restart the display of messages.

[My procedure produces unexpected results when I use one of the Find methods.](#)

The **FindFirst**, **FindLast**, **FindNext**, and **FindPrevious** methods might produce unexpected results for the following reasons:

- The *criteria* argument you specified uses invalid syntax to refer to the value of a control or property.
- The *criteria* argument you specified uses invalid syntax to combine the value of a field, control, or property with a literal string.

The **FindRecord** and **FindNext** methods might produce unexpected results if the **FindRecord** method's *onlycurrentfield* argument is set to **acCurrent**. Use the **GoToControl** method to move the focus to the control containing the data you're searching for before using the **FindRecord** or **FindNext** method.

[I get a "Sub or Function not defined" message when I try to use the IsLoaded function.](#)

The IsLoaded function is a sample **Function procedure** included with the Northwind sample database, and not a built-in Microsoft Visual Basic function. To use the IsLoaded function in your database, you must copy it to your database.

1. Open the Northwind sample database.

[How?](#)

1. On the **Help** menu, point to **Sample Databases**, and then click the database you want to open.
2. If the database isn't currently installed, click **Yes** to install it now.

Note If you don't see a list of databases when you point to **Sample Databases**, you can install them by double-clicking **Add/Remove Programs** in Windows Control Panel and running Office Setup.

2. In the Database window of Northwind, click **Modules**  under **Objects**.
3. Select **Utility Functions**, and then click the **Design** on the Database window toolbar.
4. Highlight and copy the function.
5. Close the Northwind sample database.
6. In the Database window of your database, click **Modules**  under **Objects** and then click **New** on the Database window toolbar.
7. Paste the code into the module.
8. Close and save the new module.

Microsoft Access displays this message if you use the . (dot) operator rather than the ! operator to refer directly to a table field. For example, this error is produced by the last line of the following block of [Microsoft Visual Basic](#) code:

```
Dim dbsCurrent As Database
Dim tblTest As TableDef
Dim fldTest As Field

Set dbsCurrent = CurrentDb
Set tblTest = dbsCurrent.TableDefs("Categories")
Set fldTest = tblTest.CategoryID
```

To correct the problem, replace the . (dot) operator with the ! operator or refer to the field as an element of the **Fields** collection. You can use either of the following lines of code, although the first might result in faster performance:

```
Set fldTest = tblTest!CategoryID
```

```
Set fldTest = tblTest.Fields("CategoryID")
```

[Show All](#)

Create a message box programmatically

When you want to display a brief message, such as a warning, you can use a predefined dialog box called a message box. You can display a message box by adding the MsgBox action to a [macro](#) or by adding the **MsgBox** function to a [Microsoft Visual Basic procedure](#).

[Use a macro](#)

1. Create a macro.

[How?](#)

1. In the [Database window](#), click **Macros**  under **Objects**.
2. Click the **New** button on the Database window toolbar.
2. In a blank action row, click **MsgBox** in the [action list](#), and then set the [action arguments](#).
3. Click **Save**  to save the macro.
4. To test the macro, click **Run**  on the [toolbar](#).

[Use Visual Basic code](#)

1. Create a Visual Basic procedure.

[How?](#)

1. In the Database window, click **Modules**  under **Objects**, and then click **New** on the Database window toolbar.
2. Declare the procedure by typing the **Sub** statement.
3. Type a procedure name, immediately followed by any [arguments](#) in parentheses. For example, the following declaration for the ShowEvent [Sub procedure](#) specifies *EventName* as an argument:

```
Sub ShowEvent(EventName As String)
```

2. Add an assignment statement to the procedure that assigns the value returned by the **MsgBox** function to a [variable](#). The function returns different values depending on the button a user clicks in the message box. Your procedure can then perform different operations based on the value returned.

For example, the following assignment statement runs the **MsgBox** function, displaying a message, and assigns the value returned by the function to the variable RetValue:

```
RetValue = MsgBox("Continue?", vbOKCancel)
```

3. To test, run the procedure by clicking **Run Sub/UserForm**  on the toolbar, select the procedure in the **Macros** dialog box, and then click **Run**.

[Show All](#)

Navigate between controls, records, and form pages programmatically

You can automatically move to a control, record, or form page in response to an event that happens on a form by carrying out the **GoToControl**, **GoToRecord**, or **GoToPage** actions in a [macro](#) or an [event procedure](#).

 [Use a macro](#)

1. To create a macro, click **Macros**  under **Objects**, and then click the **New** button on the Database window toolbar.
2. Do one of the following:
 - To move the [focus](#) to a specific control or field in the current record, click **GoToControl** in the action list of a blank action row, and then set the **Control Name** argument to the name of the control to which you want to move.
 - To make a specific record the current record, click **GoToRecord** in the action list of a blank action row, and then set the arguments.
 - To move the focus in the active form to the first control on a specific form page, click **GoToPage** in the action list of a blank action row, and then set the arguments.
3. Click **Save** .
4. To test, run the macro by clicking **Run**  on the toolbar.
5. Open the form in [Design view](#).
6. Open the property sheet for the form, select the event property that you will use as a trigger and set it to the name of the macro.

For example, if a macro containing the **GoToControl** action is located in the **OnCurrent** event for the form, moving from one record to another will move the focus to the control.

Use Visual Basic code

1. To open the event procedure for the appropriate event, open the form in [Design view](#).
2. Display the property sheet for the form or control on the form and then click the **Event** tab.
3. Click the event property for the event that you want to trigger the procedure.

For example, to respond to a mouse click on a command button, open the button's OnClick event procedure.

4. Click **Build**  next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Code Builder** to display the event procedure window.
6. Do one of the following:
 - To move the focus to a specific control or field in the current record, carry out the **GoToControl** method in the [procedure](#), setting the **Control Name** argument to the name of the control to which you want to move.
 - To make a specific record the current record or to create a new record, use the **GoToRecord** method to carry out the GoToRecord action in the procedure.
 - To move the focus in the active form to the first control on a specific form page, use the **GoToPage** method to carry out the GoToPage action in the procedure.
7. To test, run the procedure by opening the form in Form view and performing the event that triggers the move.

For example, if the procedure is located in the event procedure for a command button, click the command button to observe the focus move to the control, record, or form page.

Tip

You can also use [Microsoft Visual Basic for Applications](#) to locate or move between records by using a **Recordset** object and the **Seek** method or any of the

Move or **Find** methods.

[Show All](#)

Save a database object programmatically

You can automate saving a [database object](#) by carrying out the Save action in a [macro](#) or an [event procedure](#).

[Save a database object by using a macro](#)

1. To create a macro, click **Macros**  under **Objects**, and then click the **New** button on the Database window [toolbar](#).
2. In a blank action row, click **Save** in the [action list](#), and then set the **Object Type** argument to the type of the object you want to save and the **Object Name** argument to the name of the object.
3. Click **Save** to save the macro.
4. To test, select the macro and click **Run**  on the toolbar.
5. The macro is now ready for use by setting the property of another object.

For example, to save a form by clicking a command button on the form, set the button's **OnClick** property to the name of the macro.

[Save a database object by using an event procedure](#)

1. To open the event procedure for the appropriate event, open the form in [Design view](#).
2. Display the property sheet for the control you will use to trigger the event and then click the **Event** tab.
3. Click the property for the event that you want to trigger the procedure.

For example, to respond to a mouse click on a command button, open the button's OnClick event procedure.

1. Click **Build** next to the property box to display the **Choose Builder** dialog box.
2. Double-click **Code Builder** to display the event procedure window.

3. Use the **Save** method to carry out the Save action in the procedure, setting the *objecttype* argument to the type of the object you want to save and the *objectname* argument to the name of the object.
4. To test, run the procedure by opening the form in Form view and performing the event that triggers the Save action.

For example, if the procedure is located in the OnClick event procedure for a command button, click the command button to save the object.

[Show All](#)

Show or hide warning messages programmatically

You can temporarily disable the [modal](#) warning messages displayed by Microsoft Access while a macro or Microsoft Visual Basic procedure is running, and enable them again by carrying out the SetWarnings action in the macro or procedure.

[Use a macro](#)

1. To open an existing macro, click **Macros**  under **Objects**, select the macro, and then click the **Design** button on the Database window toolbar.
2. In the place in the macro where you want to stop or resume the display of modal warning messages, add the SetWarnings action.

[How?](#)

1. In the Macro window, click the first empty row in the **Action** column. If you want to insert an action between two action rows, click the selector for the action row just below the row where you want to insert the new action, and then click **Insert Row**  on the toolbar.
 2. In the **Action** column, click the arrow to display the action list.
 3. Click the action you want to use.
 4. In the lower part of the window, specify arguments for the action, if any are required. For action arguments whose settings are a database object name, you can set the argument by dragging the object from the Database window to the action's **Object Name** argument box.
 5. Type a comment for the action. Comments are optional.
3. To temporarily stop the display of warning messages, set the **Warnings On** argument to **No**. To resume displaying them, set the **Warnings On** argument to **Yes**.
 4. Click **Save**  to save the macro.
 5. To test, click **Run**  on the toolbar.

If the Warnings On argument was set to No, the display of warning messages

automatically resumes when the macro stops running.

[Use Visual Basic code](#)

1. Open a **Function** or **Sub** procedure in the Code window.
2. In the place in the procedure where you want to stop or resume the display of modal warning messages, use the **SetWarnings** method to carry out the SetWarnings action.

[How?](#)

Microsoft Access defines a special object, **DoCmd**, that you use to carry out macro actions in Microsoft Visual Basic procedures. You carry out an action by adding a method of the **DoCmd** object to your procedure.

- In the procedure, add the **DoCmd** method that corresponds to the action by using the following syntax:

DoCmd.method [*arguments*]

Method is the name of a method. *Arguments* are the method arguments, if there are any.

For example:

```
DoCmd.SetWarnings False
```

3. To temporarily stop the display of warning messages, set the *warningson* argument to **False**. To resume displaying them, set the *warningson* argument to **True**.
4. To test, run the procedure by opening the form in Form view and performing the event that triggers the procedure

If the the *warningson* argument was set to **False**, the display of warning messages automatically resumes when the code stops running.

[Show All](#)

Synchronize records between two forms programmatically

When you move between records on a form, you can display related records on a second form by running a [macro](#) or an [event procedure](#) in response to the first form's **Current** event.

[Use a macro](#)

1. To create a macro, click **Macros**  under **Objects**, and then click the **New** button on the Database window toolbar.
2. In a blank action row, click **OpenForm** in the [action list](#).
3. Set the **Form Name** argument to the name of the second form, and then set the **Where Condition** argument to display the related records you want in the second form.
4. Click **Save**  and close the macro window.
5. Open the first form in [Design view](#) and set the **OnCurrent** event property of the form to the name of the macro.
6. To test, open the first form in Form view and move from one record to another record.

The second form should open and display related records.

[Use Microsoft Visual Basic code](#)

1. Open the first form in [Design view](#).
2. To open the form's Form_Current event procedure for the appropriate event, display the property sheet for the form, and then click the **Event** tab.
3. Click the **OnCurrent** event.
4. Click **Build** next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Code Builder** to display the event procedure window.
6. Add [Visual Basic](#) code to the event procedure to display records in the

second form that have field values that match a related field on the first form.

7. To test, open the first form in Form view and move from one record to another record.

The second form should open and display related records.



You can also use the Form Wizard to create two related forms from scratch by using the **Linked Forms** option.

[Show All](#)

View the latest records for a form programmatically

You can automatically view the most current data for a form by refreshing and requerying the records.

If you're using a database that's shared on a network, you can view the latest version of existing records by creating a command button that updates form data.

To display the latest set of records in a form, including new records, you can requery the records by adding the Requery action to a [macro](#) or by adding the **Requery** method to a [Microsoft Visual Basic procedure](#).

[Use a macro](#)

1. To create a macro, open the form in [Design view](#).
2. Display the property sheet for the form or control on the form and then click the **Event** tab.
3. Click the event property for the event that you want to trigger the macro.

For example, to respond to a mouse click on a command button, open the button's OnClick event procedure.

4. Click **Build** next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Macro Builder** to display the macro window.
6. Click **Save**  to save the macro.
7. In a blank action row, click **Requery** in the [action list](#).

To update records for a [control](#) that is based on a table or query, set the **Control Name** argument to the name of the control.

To update the records for the form as a whole, leave the **Control Name** argument blank.

8. To test, run the macro by opening the form in Form view and performing the event that triggers the requery.

[Use Visual Basic code](#)

1. To open the event procedure for the appropriate event, open a form in [Design view](#).
2. Display the property sheet for the form or control on the form and then click the **Event** tab.
3. Click the event property for the event that you want to trigger the procedure.

For example, to respond to a mouse click on a command button, open the button's OnClick event procedure.

4. Click **Build** next to the property box to display the **Choose Builder** dialog box.
5. Double-click **Code Builder** to display the event procedure window.
6. Add the **Requery** method to the procedure for the form or [control](#) whose records you want to update. For example, to view the latest records in a combo box called CategoryID, add the following method to the procedure:

```
CategoryID.Requery
```

7. To test, run the procedure by opening the form in Form view and performing the event that triggers the requery.

Enable editing for multiple languages

1. On the Microsoft Windows **Start** menu, point to **Programs**, point to **Microsoft Office**, point to **Microsoft Office Tools**, and then click **Microsoft Office 2003 Language Settings**.
2. Click the **Enabled Languages** tab.
3. In the **Available languages** box, select the language you want to enable.
4. Click **Add**.

If any Microsoft Office programs are running, the Language Settings tool notifies you and closes them automatically. The features for the language you enabled appear the next time you start an Office program.

Note Some languages— for example, Asian, right-to-left (such as Arabic), and Central European languages— have system requirements that must be met before you can type characters for those languages in Microsoft Office programs. If you see **(limited support)** next to a language in the **Available languages** box, you need to install system support for that language.

[Show All](#)

About Japanese sort order

[Japanese characters are sorted in the following order](#)

- Hiragana and katakana are sorted in the order of the kana syllabary.
- English letters are sorted alphabetically. No distinction is made between uppercase and lowercase letters.
- Numbers are sorted in descending order.
- No distinction is made between half-pitch and full-pitch English characters and symbols.
- Repetition marks "" are treated the same as the character preceding the mark. For example, "" is treated as "".
- Other kanji, symbols, and so on are sorted in the order of their Shift-JIS codes.

[The order of the kana syllabary is as follows](#)

The rules are applied in this order:

1. No distinction is made between hiragana, full-pitch katakana, and half-pitch katakana.
2. Voiced sounds, semi-voiced sounds, double consonants, and contracted sounds are treated as unvoiced sounds. Long vowel sounds () are treated as the vowel of the previous syllable. For example:
 - Voiced sound: ->
 - Semi-voiced sound: ->
 - Double consonant: ->
 - Contracted sound: ->
 - Long-vowel sound: ->
3. Voiced and unvoiced sounds are sorted in the following order: unvoiced sound, voiced sound, semi-voiced sound. Previous sounds have priority.

Example: -> ->

4. Double consonants and contracted sounds are placed in front of unvoiced sounds. Previous sounds have priority.

Example: ->

[Show All](#)

Specify IME behavior

You can set properties that control [Input Method Editor](#) behavior for a field in a table or a control on a form. The **IMEMode** property applies to all East Asian languages, and the **IMEHold** and **IMESentenceMode** properties apply to Japanese only.

[Set IME properties for a field in a table](#)

You can set IME properties for a [Text](#), [Memo](#), or [Hyperlink](#) field.

1. Open the table in [Design view](#).
2. In the upper portion of the window, click the field for which you want to specify the IME behavior.
3. In the lower portion of the window, set the **IMEMode** and **IMESentenceMode** properties. For more information, click the applicable property box and press F1.

[Set IME properties for a control on a form](#)

You can set IME properties for a text box, a list box, or a combo box.

1. Open the form in [Design view](#).
2. Select the control for which you want to set the IME mode, and then click **Properties**  on the **Form Design** toolbar.
3. Set the **IMEHold**, **IMEMode**, or **IMESentenceMode** properties. For more information, click the applicable property box and press F1.

[Specify IME behavior when entering data in a datasheet](#)

1. On the **Tools** menu, click **Options**, and then click the **Keyboard** tab.
2. Do one of the following:
 - To specify that the **IMEMode** property is set to **No Control** when entering data in a datasheet, select the **Datasheet IME control** check box.
 - To specify that IME behavior is determined by the **IMEMode** property

of the individual fields, clear the **Datasheet IME control** check box.

[Show All](#)

Convert Hangul and Hanja

If you are not using the Korean language version of Microsoft Office, then this feature is only available if you have installed the Microsoft Office 2003 Multilingual User Interface Pack or Microsoft Office 2003 Proofing Tools for Korean.

You can convert Hangul to Hanja and vice-versa. For example, convert the Hangul word "학교" to the Hanja word "學校". You can also set conversion options.

[Convert text from Hangul to Hanja or vice-versa](#)

1. Type the text you want to convert, and then select it.
2. Click the **Language**  icon on the taskbar, and click **Show the Language Bar**. On the language bar, select **Korean**. Click the **Hanja** icon.
3. In the **Hanja Suggestions** or **Hangul Suggestions** box, select the appropriate converted text.
4. Do one of the following:
 - To convert the selected word in the **Suggesting For** box to the word in the **Hanja Suggestions** or **Hangul Suggestions** box, click **Change**.
 - To skip converting the selected word, click **Ignore**. If you don't want to convert the word in the rest of the document, click **Ignore All**.

[Set Hangul/Hanja conversion options](#)

1. Select the sentence containing Hangul or Hanja.
2. Click the **Language**  icon on the taskbar, and click **Show the Language Bar**. On the language bar, select **Korean**. Click the **Hanja** icon.
3. Click **More**, click **Options**, and then select the options you want.
 - With **Ignore Hangul ending** selected, the ending will not be converted.

- With **Display Recently Used Items** selected, the most recently used word will be displayed at the top of the **Suggestions** list.
- You can also set the conversion type and display type.

[Show All](#)

Add or delete Hanja in a custom dictionary

If you are not using the Korean language version of Microsoft Office, then this feature is only available if you have installed the Microsoft Office 2003 Multilingual User Interface Pack or Microsoft Office 2003 Proofing Tools for Korean. In addition, you may need to enable support for Korean through Microsoft Office 2003 Language Settings.

[Add Hanja to a custom dictionary](#)

1. Type the Hangul for the Hanja you want to add, and then select it.
2. Click the **Language**  icon on the taskbar, and click **Show the Language Bar**. On the language bar, select **Korean**. Click the **Hanja** icon.
3. Click **More**.
4. Click **Add new word**.
5. In the **Hanja Suggestions** box, select the appropriate Hanja, and then click **Choose**.
6. Click **Add to List** after all the characters are converted.

[Delete Hanja from a custom dictionary](#)

1. Type the Hangul for the Hanja you want to delete, and then select it.
2. Click the **Language**  icon on the taskbar, and click **Show the Language Bar**. On the language bar, select **Korean**. Click the **Hanja** icon.
3. In the **Hanja Suggestions** box, select the Hanja you want to delete.
4. Click **More**.
5. Click **Remove word**.

[Show All](#)

About Correct Keyboard Setting

Correct Keyboard Setting is a feature that automatically corrects words entered in English mode that were meant to be entered in Hangul mode, and vice versa. For example, if you enter **eogksalsrnr** in English mode, it will be changed to 대한민국; and if you enter **ㅋ ㅌ ㅍ ㅊ ㄱ 나 쇼** in Hangul mode, it will be changed to **university**. Also, [IME](#) mode is changed automatically.

Correct Keyboard Setting also automatically corrects English words that have a Hangul ending entered in English mode, and vice versa. For example, if you enter **optiondmf** in English mode, it is automatically changed to **option**를.

Correct Keyboard Setting works even when Hangul and English are mixed. For example, if you enter **rntjddnjs**버미ㅏ쇼, it is automatically changed to 구성원 **quality**.

Turn on Correct Keyboard Setting

To use this procedure, you must have a U.S. English version of the Microsoft Windows operating system.

1. On the **Tools** menu, click **AutoCorrect Options**.
2. On the **AutoCorrect** tab, select the **Correct Keyboard Setting** check box.

[Show All](#)

Add a word to an Input Method Editor (IME) dictionary

Note The feature described in this Help topic is only available if an [Input Method Editor \(IME\)](#) for Japanese, Simplified Chinese or Traditional Chinese has been installed.

[Add a word to the Japanese IME dictionary](#)

1. Click the **Language** icon  on the taskbar, and click **Show the Language bar**. On the language bar, select **Japanese**. Click the **Tools** button.
2. On the **Tools** menu, click **Dictionary Tool**.
3. On the **Edit** menu of the **Microsoft IME Dictionary Tool** dialog box, click **Add**.
4. Click on the **Add Word** tab.
5. In the **Reading** box, enter the reading of the word you want to register in hiragana or alphanumeric characters.
6. In the **Display** box, enter the display of the word you want to register.
7. Click **Register** to save the word.

[Add a phrase to the Simplified Chinese dictionary](#)

1. Click the **Language** icon  on the taskbar, and click **Show the Language bar**. On the language bar, select **Chinese (Taiwan)**. Click the **Tools** button.
2. On the **Tools** menu, click **User phrase**.
3. On the **Edit** menu of the **Microsoft New Phonetic IME 2002 user define phrase tool** dialog box, click **Add Phrase**.
4. Type the phrase in the **Phrase** box, and then press ENTER.
5. On the **File** menu, click **Save**.

[Add a phrase to the Traditional Chinese dictionary](#)

1. Click the **Language** icon  on the taskbar, and click **Show the Language bar**. On the language bar, select **Chinese (PRC)**. Click the **Context menu**

button.

2. On the **Context menu**, click **User-defined phrase tool**.
3. On the **Edit** menu of the **MSPY IME User-defined Phrase Tool**, click **Add Phrase**.
4. Type the phrase in the **Phrase** box, and the press ENTER.
5. On the **File** menu, click **Save**.

[Show All](#)

Work with euro currency symbols

Microsoft Access 2000 (and later versions) provides full support for displaying, entering, and printing the euro currency symbol €.

 [Typing the euro currency symbol](#)

To enter the euro currency symbol €, use the following keys:

Press	On
ALT+0128 (numeric keypad)	Arabic, Hebrew, Turkish, and United States 101 keyboards
ALTGR+E	Belgian, Belgian Dutch 120, Croatian, Czech, Czech 101, Czech Programmer's, Danish, Dutch KBD143, Estonian, Faroese, Finnish, French, German (IBM), German (Standard), Icelandic, Italian, Italian 142, Latvian, Lithuanian, Lithuanian (New), FYRO Macedonian (Cyrillic), Norwegian, Portuguese-KBD163, Serbian (Cyrillic), Serbian (Latin), Slovak, Slovak-QWERTY, Slovenian, Spanish, Swedish, Swiss French, Swiss German, Turkish F 440, and Turkish Q 179 keyboards
ALTGR+epsilon	Greek keyboards
ALTGR+4	Irish, Latvian-QWERTY, and United Kingdom keyboards
ALTGR+5	Greek Latin and U.S.-International keyboards
ALTGR+U	Hungarian, Hungarian 101, Polish, and Polish Programmer's keyboards
ALT+0136 (numeric keypad)	Cyrillic keyboards

 [Printing the euro currency symbol](#)

If your printer does not have the euro currency symbol in any of its resident fonts, a box will be printed instead of the euro currency symbol. Contact your printer vendor to find out how to get updated printer fonts that include the euro

currency symbol.

About South Asian databases

All South Asian databases in Microsoft Access share the following feature.

File compatibility Access can open databases created in earlier versions of Access; however, in order to display South Asian text correctly, the applicable South Asian language must be enabled.

Note Databases converted from Access 97 that have Thai date formats may not display correctly in Access 2000 or later versions.

[Show All](#)

About using numbers and dates in South Asian databases

South Asian numeral formats are used in Microsoft Access 2000 and later version databases as numbers, and in date formats. Since Vietnamese uses Arabic numerals and the [Gregorian calendar](#), there are no special requirements for calculating numbers and only a few differences in displaying formats from what English users would expect— for example, in Vietnamese the currency symbol appears after numbers, commas are used in place of periods in decimal places, and periods are used instead of commas as thousands separators. However, in Thai, Access provides a property that lets you choose whether you want Arabic or Thai numerals.

Numeral Shapes property (Thai only)

Two different representations of digit shapes are used in Thai: Arabic and Thai numerals. You can choose to have the text in controls be specifically Arabic or Thai, or you can let the Microsoft Windows operating system determine the numerals. For more sophisticated error handling, refer to the discussion of the Microsoft Visual Basic for Applications (VBA) **Numeral Shapes** property.

Dual calendars

The date format used by Access is determined by your system locale setting — for example, to display dates in Thai date format, Thai must be selected as your locale in the regional settings of Windows **Control Panel**. You can enter date values in both Gregorian and Buddhist calendars in the same database by specifying the applicable calendar.

[Show All](#)

Specify the initial keyboard language in a text box or combo box

You can select the type of [keyboard language](#) you want in a text box or combo box.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the toolbar.
3. On the **All** tab, in the **Keyboard Language** box, do one of the following:
 - To make the keyboard language follow the most recent keyboard language you used, click **System**.
 - To make the keyboard language a particular language, click one of the languages shown in the list.

Note The languages that appear in the **Keyboard Language** property box are determined by the keyboard languages that you have installed by double-clicking **Keyboard** in Microsoft Windows **Control Panel**.

[Show All](#)

Specify the appearance of numbers in a text box or other control

You can set the appearance of numbers in a label, text box, combo box, or list box.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the toolbar.
3. On the **All** tab, in the **Numerical Shapes** box, do one of the following:
 - For numerals formatted according to the operating system setting for default keyboard, click **System** (for Microsoft Windows 2000, see the **Input Locales** tab under **Regional Options** in the **Control Panel**).
 - For Arabic numerals, click **System**.
 - For Thai numerals, click **National**.

[Show All](#)

Specify dates in dual calendar databases

- Do one of the following:
 - To convert to Buddhist calendar dates when the regional settings in Microsoft Windows **Control Panel** are set to the [Gregorian calendar](#), type the letters **B** or **b** in front of the year entry in a date format— for example, to enter the Buddhist date 1/1/42 (or January 1, B.E. 2542), type **1/1/b42** and Microsoft Access displays its Gregorian equivalent, 1/1/1999.
 - To convert Gregorian calendar dates when the regional settings in Windows **Control Panel** are set to the Buddhist calendar, type the letters **G** or **g** in front of the year entry in a date format.

Note This technique applies only when you are using a date as criteria in a query.

[Show All](#)

Justify Thai text

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the toolbar.
3. On the **All** tab, in the **Text Align** box, select **Distribute** from the list.

Note When text displays in a control in a multi-line format, each line of text is distributed independently according to the number of characters in the line.

About SQL Server 7.0 Support

When using Access in Client Server mode you must have SQL Server 7.0 installed with a Thai-enabled character set to display Thai characters and correctly sort and search. Choose from the following:

This character set	Supports
Thai (874)	Thai and English only
Multilingual (850)	Thai and other languages simultaneously

[Show All](#)

About right-to-left behavior in forms and reports

Right-to-left characteristics for forms and reports are summarized as follows:

[Default direction](#)

The default [direction](#) that was set on the **International** tab in the **Options** dialog box (**Tools** menu) determines whether your form has a [left-to-right](#) or right-to-left layout. If you have not changed the default, new controls added to the form are displayed with a right-to-left direction.

[Ruler and grid](#)

Whether a form or report is in a right-to-left or left-to-right direction, the ruler retains its origin in the upper-left corner of the design window. The same concept applies to the grid. This provision does not cause any problems when creating a right-to-left-oriented form. By dragging the right grid boundary to the right, controls on the grid move to the right in proportion to the distance the boundary is moved and the grid is extended from the left boundary.

[Set control defaults](#)

The **Set Control Defaults** command on the **Format** menu recognizes all right-to-left properties in Microsoft Access 2000 (and later versions). When you select this command, Access applies all right-to-left properties.

[Change To](#)

When you change a control's type (for example, from a text box to a list box) by using the **Change To** command, Access transfers any relevant right-to-left properties from the original control to the new one.

[Snap To Grid and Align To Grid](#)

These **Format** menu commands behave according to the direction of the current form or report.

Snap To Grid When you place a new control on a left-to-right form or report, the upper-left corner of the control is aligned to the nearest point on the grid. When you place a new control on a right-to-left form or report, the upper-right corner aligns to the nearest point.

Align To Grid In a left-to-right form or report, the upper-left corners of selected controls align to the nearest point in the grid. In a right-to-left form or report, upper-right corners of selected controls align to the nearest grid point.

 [Form Datasheet view](#)

Datasheet view in forms is the way to present tables with the most sophisticated right-to-left behavior. In addition to displaying the underlying data in the column and row layout generally associated with tables, each field in Datasheet view has the right-to-left properties of the associated Form view control.

For example, if you set the **Numeral Shapes** property to Arabic for a text box control bound to a Text field named Sales, then the [keyboard language](#) will automatically switch to Arabic whenever you click in the Sales field in Datasheet view. This feature is useful for volume data-entry applications because it reduces the need to make keyboard language changes when moving from one field to the next.

 [Form and report Design view](#)

A hierarchy of control governs the orientation, overall appearance, and basic parameters of forms and reports. The following progression illustrates this point:

1. The **Default Direction** setting on the **International** tab in the **Options** dialog box (**Tools** menu) sets the text direction for new forms or reports to right-to-left (unless the default was manually changed).
2. In the property sheet for a form or report, you can change the direction for a specific form or report by setting the **Orientation** property. This change will determine the default value of related left-to-right and right-to-left direction properties for each new control on that form or report.
3. Each control has a default set of right-to-left properties that can be changed

for that individual control.

4. After controls are placed on the design grid, you can move them and change their right-to-left properties to change their appearance and behavior.



[Print Preview and Layout Preview in reports](#)

Access displays the report in the direction in which it was designed, left-to-right or right-to-left.

[Show All](#)

About right-to-left feature compatibility

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Access 2000 (and later versions) provides full forward compatibility for database files saved in all previous versions of Access with right-to-left support.

[Show All](#)

About right-to-left language features

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

When languages such as Arabic, Farsi, Hebrew, and Urdu are enabled for editing, you have several features available to you to make working in a right-to-left or [bidirectional](#) environment easier.

[About bidirectional fonts](#)

Access is designed to display and print both right-to-left and [left-to-right](#) text. For best results, the program should use a font that contains both right-to-left and left-to-right language characters. Some fonts available with Office and bidirectional Windows operating systems contain both right-to-left and left-to-right characters.

If you choose a font that does not contain both right-to-left and left-to-right characters, existing text in one or both languages may be displayed using a secondary font, such as Tahoma. For further information, refer to 'Unicode support' in Microsoft Office Help.

[About checking the spelling of bidirectional text](#)

You can check the spelling of both your English and right-to-left text in the Memo, Text, and Hyperlink fields. Just choose a dictionary that corresponds to the text that will be used.

When right-to-left text is checked, English words are ignored and not flagged as misspelled. Similarly, when English text is checked, right-to-left words are ignored.

[About using bidirectional text in expressions](#)

The feature described in this Help topic is only available if support for [right-to-left](#) languages is [enabled](#) through Microsoft Office Language Settings.

In addition, to enable the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, Microsoft Windows 2000.

Access preserves expressions with left alignment and [left-to-right reading order](#). Expression terms are preserved in left-to-right order to ensure the visual readability of expression statements that contain mixed text. As each statement character is entered, the expression display is dynamically updated.

The following table provides examples of [bidirectional](#) text that is used in expressions.

Expressions	Description
=Left([الولاية الأولى], 4)	Extracts the first four characters in the field الولاية الأولى (characters are extracted in logical order).
= "City:" & [المدينة]	Prints the Latin word City: followed by the value for each entry in the field المدينة.
=Sum([التكلفة]*1.1)	Multiplies each value in the field التكلفة by 1.1 and sums the results.

There are a few other considerations when using expressions with right-to-left languages:

- Access displays [Structured Query Language \(SQL\)](#) statements by using the same rules it uses for displaying expressions.
- Bidirectional [control characters](#) entered in expressions are removed prior to evaluating the expression to avoid possible conflicts.

Note Bidirectional text used in the Expression Builder will "swap" certain [neutral](#) character pairs such as "(" and "{" }" when the [keyboard language](#) is a right-to-left language. To avoid this, change the keyboard language to a left-to-right language or type the correct character.

[About right-to-left features compatibility](#)

Access 2000 (and later versions) provides full forward compatibility for database files saved in all previous versions of Access with right-to-left support.

[About using different calendars](#)

Access supports use of the Hijri calendar in addition to the standard Gregorian calendar in all locales except Thai and East Asian locales.

If you open an existing database with date references and you have selected the wrong calendar setting, the program might report a variety of errors or other unexpected behaviors. For example, if you have checked the **Hijri Calendar** setting, Access will display a syntax error message if a report makes reference to [Gregorian](#) dates that are invalid as [Hijri](#) dates (for example, the date 12/31/99 will produce a syntax error because no Hijri month has 31 days).

To avoid problems of this sort, all date references in a database must be based on the same calendar.

Important It is recommended that you choose the database calendar when you first create your database, and then never change it.

Other considerations when the database calendar is set to Hijri

- To ensure Hijri dates strings are evaluated correctly in expressions and Visual Basic for Applications statements, use the **CDate** function to evaluate the strings instead of declaring the strings as constants.

For example, type **CDate("1/25/20")** instead of **#1/25/20#**.

The **CDate** function will always return a value based on the active database calendar, whereas date constants are always evaluated by using the Gregorian calendar.

- For reliable behavior, enter and display dates in an unambiguous format. For example, dates in short date format might be misinterpreted in queries if the year or the day of the month are 12 or less (for example, 3/11/10). Dates in medium date format display only the first few characters of the month name, which can create ambiguity or an undesirable appearance. Use long date format whenever possible. You can alter the way a date is displayed in long format (for example, excluding the day of the month) by using the regional settings in Microsoft Windows **Control Panel**.
- Dates entered in the **ValidationRule** property of a Date/Time field in a table (in table [Design view](#)) must be Hijri. However, to have the **ValidationRule** date appear as a Hijri date in validation messages, the validation rule must use the **CDate** function (for example, 'CDate ("1/25/1420")').
- Sample Access applications, such as the Northwind database, are not localized and expect the database calendar to be Gregorian. Because the sample applications contain hard-coded date constants that are not valid Hijri dates, running the sample applications with the database calendar set to Hijri is not recommended.
- All Hijri years will be displayed as four-digit year values (for example, 1420) unless the **Format()** function is explicitly used.
- The Visual Basic **Date** function, used to set the system date for your computer, will not accept Hijri dates. If the database calendar is set to Hijri, set the calendar to Gregorian before setting system date with the **Date** function.

- Dates in tables exported as text files are always stored as Gregorian values. If the database calendar is set to Hijri, automatic Hijri to Gregorian conversion is performed during the export process. Similarly, dates in tables imported from text files are assumed to be Gregorian dates and will be converted to Hijri values during the import process if the database calendar setting is Hijri.

[About right-to-left sort orders](#)

The default Access sort order, [General](#), handles sorting of right-to-left characters; that is, there are no separate sort orders for right-to-left languages. General sort order now includes right-to-left characters because characters are stored as [Unicode](#) values. The behavior of General sort order for right-to-left characters stored as Unicode values is identical to the behavior of right-to-left characters previously stored as Arabic or Hebrew code page values and sorted by using Arabic sort order and Hebrew sort order, respectively.

Right-to-left text is sorted along the following guidelines:

- [Left-to-right](#) text is always sorted ahead of right-to-left text
- Sort ignores [kashidas](#) and [bidirectional control characters](#)
- Characters with [diacritics](#) are clustered after other occurrences of the base character

Note When opened or imported, previous-version databases that use a right-to-left sort order are saved with the Access General sort order.

[About importing, exporting, and linking right-to-left tables in text formats](#)

Right-to-left enabled Access supports several code pages used in importing, saving, and exporting HTML Web pages. See the next topic *About code page and text layout support for right-to-left HTML files* for more information about supported code pages

A table or worksheet in Lotus, dBASE, Microsoft FoxPro, Paradox, or ODBC database format that requires code page or text layout conversion can still be imported into Access by saving the table or worksheet in text file format prior to importing it. The ability to save a table or worksheet in text file format is a standard feature of most database or spreadsheet products. When Access imports

the text file, all field names and records from the original table or worksheet will be fully recreated in a new Access table.

To import, export, or link text files that are not recognized by Access, use ConvText.exe, a stand-alone text-file conversion utility that provides both logical and [visual](#) text layout options and an expanded list of supported [bidirectional](#) code pages. ConvText.exe is available on the Microsoft Office Web site and on the installation CDs for Microsoft Office language versions that offer right-to-left support.

[About code page and text layout support for right-to-left HTML files](#)

Right-to-left enabled Access supports several code pages used in importing, saving, and exporting HTML Web pages.

Code page conversion of tables or worksheets with right-to-left characteristics is provided for HTML files stored in [logical](#) text layout and one of the following right-to-left code pages:

Language	Code page
Arabic	UTF-8 (default)
	ISO 8859-6
	DOS-720
	ASMO-708
Hebrew	UTF-8 (default)
	ISO 8859-8
	DOS-862

Note Access supports editing and saving of HTML files in logical text layout. HTML files stored in [visual](#) text layout can be viewed but not edited or saved in logical text layout.

[About creating macros in right-to-left applications](#)

The following are some considerations to keep in mind:

- You can create custom menu bars in a right-to-left or [left-to-right direction](#).
- The TransferText action allows you to choose code page format when

importing, exporting, and linking files.

- Menu bars you create can be displayed with a left-to-right or right-to-left direction. For example, if oriented right-to-left, the menu bar will display with menu items starting on the right side of the menu bar and continuing to the left in a right-to-left [reading order](#).
- The display direction of global and form menu bars is determined by the [user-interface language](#).
- You can name menu items in right-to-left, left-to-right, or [bidirectional](#) text. An ampersand (&) can be added to a menu item name to designate a menu item access key. The ampersand applies to the next character in the name and causes the character to be underlined when the menu item is displayed. In right-to-left text, this is the character to the left of the ampersand; in left-to-right text, it's the character to the right of the ampersand.
- Macro names can be in right-to-left, left-to-right, or bidirectional text.
- The Macro window has a left-to-right layout regardless of the user-interface language.



[About creating modules in right-to-left applications](#)

Access supports these right-to-left characteristics in modules:

- You can view and edit Visual Basic for Applications statements containing right-to-left or [bidirectional](#) text.
- Statements are displayed left-aligned and in [left-to-right reading order](#).
- You can use right-to-left, left-to-right, or bidirectional text in module names.

Visual Basic for Applications

All Microsoft Visual Basic for Applications procedures and properties that provide right-to-left functionality to Access are listed alphabetically in the following table.

Procedures and properties	Type
Calendar	Property
DataType	Property
Date, Date\$	Function
DateAdd	Function
DateDiff	Function
DatePart	Function
DateSerial	Function
DateValue	Function
Day	Function
Format, Format\$	Function
Format	Property
GetOption, SetOption	Methods
KeyboardLanguage	Property
LabelAlign, TextAlign	Properties
LabelX, LabelY	Properties
Month	Function
NumeralShapes	Property
Orientation	Property
ReadingOrder	Property
ScrollBarAlign	Property
TransferText	Action/Method
Weekday	Function
Year	Function

For development of sophisticated right-to-left applications, you might want to take advantage of the numerous Microsoft Windows operating system

application programming interfaces (APIs) that have right-to-left settings. For more information, you can go to the Microsoft Office Web site.

[Show All](#)

Use right-to-left features in forms and reports

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

[Set right-to-left properties in controls](#)

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

[Specify text reading order for a text box or other control](#)

You can select the type of [reading order](#) you want in a label, text box, toggle button, combo box, list box, or command button.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **ReadingOrder** box, do one of the following:
 - To have the reading order follow the language of the text first entered, click **Context**.
 - To force the reading order in one [direction](#), click **Right-To-Left** or **Left-To-Right**.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to

dynamically change reading order and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is excluded because changing alignment might negatively affect appearance). When the focus is moved outside the control, the display format returns to the previous control reading order.

[Specify the initial keyboard language in a text box or combo box](#)

You can select the type of [keyboard language](#) you want in a text box or combo box.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **KeyboardLanguage** box, do one of the following:
 - To make the keyboard language follow the most recent keyboard language you used, click **System**.
 - To make the keyboard language a particular language, click one of the languages shown in the list.

Note The languages that appear in the **KeyboardLanguage** property box are determined by the keyboard languages you have installed. Available languages can be reviewed by double-clicking **Keyboard** or **Regional Options** in Microsoft Windows **Control Panel**.

[Specify text alignment in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **TextAlign** box, click one of the following:
 - To have text alignment follow the language used when the text was first entered or the [user interface language](#), click **General**.
 - To set a fixed alignment, click one of the other choices.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to dynamically change [reading order](#) and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is not changed because changing alignment might negatively affect appearance). When the focus leaves the control, the display format returns to the previous control reading order.

[Specify where you want scroll bars in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **ScrollBarAlign** box, do one of the following:
 - To have the scroll bar follow the setting of the **Orientation** property for the form or report that contains the control, click **System**. (A vertical scroll bar is placed on the left if the form or report **Orientation** property setting is **Right-to-left** and on the right if the form or report **Orientation** property setting is **Left-to-right**.)
 - To have the scroll bar on the right side of the control, click **Right**.
 - To have the scroll bar on the left side of the control, click **Left**.

[Specify the appearance of numbers in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **NumeralShapes** box, do one of the following:
 - For numerals formatted according to [Unicode](#) rules for adjacent text, click **Context**. In general, this means numeral shapes are [Hindi](#) if surrounding text is Arabic.
 - For numerals formatted according to the **Numbers** tab in regional settings in Windows Control Panel, click **System**.
 - For Western (Arabic) numeral shapes, click **Arabic**.

- For locale-specific numeral shapes, click **National**.

 [See a summary of right-to-left properties for controls](#)

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access 2000 (and later versions), you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Most intrinsic Access controls can be configured to present right-to-left appearance and behavior. In some cases, this can be done by using standard Access control properties (for example, the **TextAlign** property) and repositioning of controls to achieve a right-to-left appearance.

However, in some cases right-to-left properties are needed to achieve proper right-to-left appearance and behavior. The following tables lists those controls for which new right-to-left properties are added:

	Label	Text box	Toggle button	Combo box	List box	Command button
TextAlign	X	X		X	X	
ReadingOrder	X	X	X	X	X	X
KeyboardLanguage		X		X		
ScrollBarAlign		X		X	X	
NumeralShapes	X	X		X	X	

 [See examples of printing orders](#)

You can view and print your data in the [direction](#), right-to-left or [left-to-right](#), that you assigned to the object.

Objects that are too long or wide to be printed on a single sheet of paper are printed on multiple sheets in an order that is determined by the view direction of the object. The following example shows the page numbering and printing order for a report that is three pages wide and requires six pages to print.

Object direction Multiple page printing order

Left-to-Right

1	2	3
4	5	6

Right-to-Left

3	2	1
6	5	4

Note Reports and labels can be printed on a page in a multiple-column format. The layout of the columns will be determined by the direction of the report.

[Show All](#)

About bidirectional fonts

Access is designed to display and print both right-to-left and [left-to-right](#) text. For best results, the program should use a font that contains both right-to-left and left-to-right language characters. Some fonts available with Office and [bidirectional](#) Windows operating systems contain both right-to-left and left-to-right characters.

If you choose a font that does not contain both right-to-left and left-to-right characters, existing text in one or both languages may be displayed using a secondary font, such as Tahoma. For further information, refer to 'Unicode support' in Microsoft Office Help.

[Show All](#)

Set the default direction for new tables, queries, forms, reports, and pages

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. In the **Default Direction** option group, do one of the following:
 - Click **Left-To-Right** to have new objects displayed in a [direction](#) familiar to [left-to-right](#) users.
 - Click **Right-To-Left** to have new objects displayed in a direction familiar to right-to-left users.

[Show All](#)

About global text direction

[Global text direction](#) defines the starting alignment, [reading order](#), and [keyboard language](#) in edit fields. However, you can change these properties by using key combinations or other methods, such as the **Formatting** toolbar. Changing the paragraph direction in any edit field affects that field only; no other edit field in the item is affected.

Global text direction also affects:

- Reading order of text displayed in the columns of all of its Table views.
- Laying out plain text information in an item's body or content area. Because plain text does not carry a paragraph direction property in it, a message received from a plain text mail client or item heading information pasted in the body of the item when replying or forwarding will have a layout that depends on the global text direction.

[Show All](#)

Change global text direction settings

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. Choose one of the following:

Click **Left-to-Right** to set edit field starting alignment to left, reading order to [left-to-right](#), and keyboard language to the currently enabled left-to-right language.

Click **Right-to-Left** to set edit field starting alignment to right, [reading order](#) to right-to-left, and [keyboard language](#) to the currently enabled right-to-left language.

Note When you change the [global text direction](#) setting, it will affect the edit fields of items opened after you change in the setting; it will not affect the text direction of the edit fields of items already open.

[Show All](#)

About right-to-left control properties

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

The settings for these properties are initially determined when the control is placed on the grid in form or report [Design view](#).

Note The settings associated with these properties are available for forms and reports. Control behavior for tables and queries is a fixed, [bidirectional](#)-capable configuration: **ScrollBarAlign** = [System](#), **NumeralShapes** = System, **KeyboardLanguage** = System, **TextAlign** = [General](#), **ReadingOrder** = [Context](#), and initial insertion point position = data dependent. If the control contains no text, then alignment, [reading order](#), and initial insertion point position are determined by the [keyboard language](#) when the control is entered: right-to-left if the keyboard language is a right-to-left language, and [left-to-right](#) if otherwise.

[ReadingOrder](#) property

Determines the reading order of the control. You can specify right-to-left or left-to-right reading order or choose to have the reading order determined by the keyboard language of text first entered.

[KeyboardLanguage](#) property

The **KeyboardLanguage** property overrides the keyboard language that is currently in use (including changing the active keyboard language icon on the taskbar). This property specifies the initial keyboard language when the control receives focus. This feature is useful for volume data entry programs because it reduces the need to make keyboard language changes when moving from one field to the next. You can subsequently choose a new keyboard language.

In [Form view](#) or [Datasheet view](#), Access will activate the designated keyboard language. If the operation fails (most likely because the requested keyboard language is not available on the developer or user system), the active keyboard language is not changed.

[TextAlign](#) property

This property determines the placement of text within a control. In addition to the standard left, right, and center character alignments, right-to-left features in Access use the General alignment setting to specify how text is aligned.

[ScrollBarAlign](#) property

This property allows you to place the control's vertical scroll bars and buttons in the applicable right-to-left or left-to-right [direction](#). To have the scroll bar follow the setting of the **Orientation** property for the form or report that contains the control, click **System**. (A vertical scroll bar is placed on the left if the form or report **Orientation** property setting is **Right-to-left**; and on the right if the form or report **Orientation** property setting is **Left-to-right**.)

[NumeralShapes](#) property

This property determines if Western (Arabic) or locale-specific shapes will be used to display numeral values. For example, two different representations of digit shapes are used in number of Arabic countries: Arabic and Hindi. You can choose to have text in controls be specifically Arabic or National (typically Hindi in an Arabic locale), you can let the Microsoft Windows operating system determine the numeral shapes to display, or you can choose to have the numeral shape be determined by the language context of adjacent text (Arabic and Hindi shapes only).

[Right-to-left properties supported by different control types](#)

[Show All](#)

About the hierarchy of setting right-to-left features

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

The appearance of an [object](#) and the behavior of its right-to-left features are determined by a hierarchy of settings. The following progression illustrates this point:

1. The **Default Direction** setting on the **International** tab in the **Options** dialog box (**Tools** menu) sets the [direction](#) for new forms or reports to right-to-left (unless the default was manually changed to [left-to-right](#)).
2. In the property sheet for a form or report, you can change the direction for a specific form or report by setting the **Orientation** property. This change will determine the default value of related left-to-right and right-to-left direction properties for each new control on that form or report.
3. Each control has a default set of right-to-left properties that can be changed for that individual control.
4. After controls are placed on the design grid, you can move them and change their right-to-left properties to change their appearance and behavior.

[Show All](#)

About bidirectional properties of the user interface

The options that appear in dialog boxes and other elements of the Access user interface do not have properties that can be set. Although you cannot change their property settings, they do have [bidirectional](#) characteristics. Access assigns the following values:

Property	Value
KeyboardLanguage	System
TextAlign	General
ReadingOrder	Context
ScrollBarAlign	System
NumericalShapes	System

If the control contains no text, [reading order](#) is determined by the [keyboard language](#) when the control is entered: Right-to-left if the keyboard language is a right-to-left language, and [left-to-right](#) if otherwise. Initial alignment and insertion point position are determined by the General alignment setting.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT keyboard sequences used to change keyboard language, reading order, and alignment in other bidirectional applications will temporarily change reading order and alignment.

[Show All](#)

Examples of printing orders

You can view and print your data in the [direction](#), right-to-left or [left-to-right](#), that you assigned to the object.

Objects that are too long or wide to be printed on a single sheet of paper are printed on multiple sheets in an order that is determined by the view direction of the object. The following example shows the page numbering and printing order for a report that is three pages wide and requires six pages to print.

Object direction Multiple page printing order

Left-to-Right	1	2	3
	4	5	6
Right-to-Left	3	2	1
	6	5	4

Note Reports and labels can be printed on a page in a multiple-column format. The layout of the columns will be determined by the direction of the report.

About checking the spelling of bidirectional text

You can check the spelling of both your English and right-to-left text in the Memo, Text, and Hyperlink fields. Just choose a dictionary that corresponds to the text that will be used.

When right-to-left text is checked, English words are ignored and not flagged as misspelled. Similarly, when English text is checked, right-to-left words are ignored.

[Show All](#)

About using right-to-left sort orders

The default Access sort order, [General](#), handles sorting of right-to-left characters; that is, there are no separate sort orders for right-to-left languages. General sort order now includes right-to-left characters because characters are stored as [Unicode](#) values. The behavior of General sort order for right-to-left characters stored as Unicode values is identical to the behavior of right-to-left characters previously stored as Arabic or Hebrew code page values and sorted by using Arabic sort order and Hebrew sort order, respectively.

Right-to-left text is sorted along the following guidelines:

- [Left-to-right](#) text is always sorted ahead of right-to-left text
- Sort ignores [kashidas](#) and [bidirectional control characters](#)
- Characters with [diacritics](#) are clustered after other occurrences of the base character

Note When opened or imported, previous-version databases that use a right-to-left sort order are saved with the Access General sort order.

[Show All](#)

Set insertion point behavior

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

1. On the **Tools** menu, click **Options** and then click the **International** tab.
2. Under **Cursor movement**, do one of the following:
 - Click **Logical** to have insertion point movement according to the [logical order](#) of mixed text.
 - Click **Visual** to have insertion point movement that ignores text language and moves contiguously to the right or left.

[Show All](#)

About code page and text layout support for right-to-left HTML files

Right-to-left enabled Access supports several code pages used in importing, saving, and exporting HTML Web pages.

Code page conversion of tables or worksheets with right-to-left characteristics is provided for HTML files stored in [logical](#) text layout and one of the following right-to-left code pages:

Language	Code page
Arabic	UTF-8 (default)
	ISO 8859-6
	DOS-720
	ASMO-708
Hebrew	UTF-8 (default)
	ISO 8859-8
	DOS-862

Note Access supports editing and saving of HTML files in logical text layout. HTML files stored in [visual](#) text layout can be viewed but not edited or saved in logical text layout.

[Show All](#)

About importing, exporting, and linking right-to-left tables in text formats

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support—for example, the Arabic version of Microsoft Windows 2000.

Access provides code page conversion of tables and worksheets with right-to-left characteristics for text files stored in [logical](#) text layout, with one of the following right-to-left code pages:

Language	Code page
Arabic	UTF-8 (default)
	ISO 8859-6
	DOS-720
	ASMO-708
Hebrew	UTF-8 (default)
	ISO 8859-8
	DOS-862

To import, export, or link text files that are not recognized by Access, use ConvText.exe, a stand-alone text-file conversion utility that provides both logical and [visual](#) text layout options and an expanded list of supported [bidirectional](#) code pages. ConvText.exe is available on the Microsoft Office Web site and on the installation CDs for Microsoft Office language versions that offer right-to-left support.

Note A table or worksheet in Lotus, dBASE, Microsoft FoxPro, Paradox, or ODBC database format that requires code page or text layout conversion can still

be imported into Access by saving the table or worksheet in text file format prior to importing it. The ability to save a table or worksheet in text file format is a standard feature of most database or spreadsheet products. When Access imports the text file, all field names and records from the original table or worksheet will be fully recreated in a new Access table.

[Show All](#)

About creating macros in right-to-left applications

The following are some considerations to keep in mind:

- You can create custom menu bars in a right-to-left or [left-to-right direction](#).
- The TransferText action allows you to choose code page format when importing, exporting, and linking files.
- Menu bars you create can be displayed with a left-to-right or right-to-left direction. For example, if oriented right-to-left, the menu bar will display with menu items starting on the right side of the menu bar and continuing to the left in a right-to-left [reading order](#).
- The display direction of global and form menu bars is determined by the [user-interface language](#).
- You can name menu items in right-to-left, left-to-right, or [bidirectional](#) text. An ampersand (&) can be added to a menu item name to designate a menu item access key. The ampersand applies to the next character in the name and causes the character to be underlined when the menu item is displayed. In right-to-left text, this is the character to the left of the ampersand; in left-to-right text, it's the character to the right of the ampersand.
- Macro names can be in right-to-left, left-to-right, or bidirectional text.
- The Macro window has a left-to-right layout regardless of the user-interface language.

[Show All](#)

Change the default for a right-to-left feature

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.



[Set the direction for new tables, queries, forms, reports, and pages](#)

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. In the **Default Direction** option group, do one of the following:
 - Click **Left-To-Right** to have new objects displayed in a [direction](#) familiar to [left-to-right](#) users.
 - Click **Right-To-Left** to have new objects displayed in a direction familiar to right-to-left users.



[Set insertion point behavior](#)

1. On the **Tools** menu, click **Options** and then click the **International** tab.
2. Under **Cursor movement**, do one of the following:
 - Click **Logical** to have insertion point movement according to the [logical order](#) of mixed text.
 - Click **Visual** to have insertion point movement that ignores text language and moves contiguously to the right or left.



[Specify alignment behavior for General alignment](#)

1. On the **Tools** menu, click **Options** , and then click the **International** tab.

2. In the **General Alignment** option group, do one of the following :
 - Click **Interface Mode** to align text and numbers in text fields according to the [user interface language](#) in effect.
 - Click **Text Mode** to align text and numbers in text fields according to the [keyboard language](#) of the text first entered.

[Switch between Gregorian and Hijri calendars for the current database](#)

Any version of Access that supports right-to-left text (except Hebrew) also supports the use of the [Gregorian](#) (default) and [Hijri](#) calendar for date entry and display.

1. Click **Options** on the **Tools** menu, and then click the **International** tab.
2. Select the **Use Hijri Calendar** check box.

The setting is saved with the database and used for all date references in the database.

[Show All](#)

About using bidirectional text in expressions

The feature described in this Help topic is only available if support for [right-to-left](#) languages is [enabled](#) through Microsoft Office Language Settings.

In addition, to enable the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, Microsoft Windows 2000.

Access preserves expressions with left alignment and [left-to-right reading order](#). Expression terms are preserved in left-to-right order to ensure the visual readability of expression statements that contain mixed text. As each statement character is entered, the expression display is dynamically updated.

The following table provides examples of [bidirectional](#) text that is used in expressions.

Expressions	Description
=Left([الطية الأولى], 4)	Extracts the first four characters in the field الطية الأولى (characters are extracted in logical order).
="City:" & [المدينة]	Prints the Latin word City: followed by the value for each entry in the field المدينة.
=Sum([التكلفة]*1.1)	Multiplies each value in the field التكلفة by 1.1 and sums the results.

There are a few other considerations when using expressions with right-to-left languages:

- Access displays [Structured Query Language \(SQL\)](#) statements by using the same rules it uses for displaying expressions.
- Bidirectional [control characters](#) entered in expressions are removed prior to evaluating the expression to avoid possible conflicts.

Note Bidirectional text used in the Expression Builder will "swap" certain [neutral](#) character pairs such as "(" and "{" when the [keyboard language](#) is a

right-to-left language. To avoid this, change the keyboard language to a left-to-right language or type the correct character.

[Show All](#)

About creating modules in right-to-left applications

Access supports these right-to-left characteristics in modules:

- You can view and edit Visual Basic for Applications statements containing right-to-left or [bidirectional](#) text.
- Statements are displayed left-aligned and in [left-to-right reading order](#).
- You can use right-to-left, left-to-right, or bidirectional text in module names.

Visual Basic for Applications

All Microsoft Visual Basic for Applications procedures and properties that provide right-to-left functionality to Access are listed alphabetically in the following table.

Procedures and properties	Type
Calendar	Property
Data Type	Property
Date, Date\$	Function
DateAdd	Function
DateDiff	Function
DatePart	Function
DateSerial	Function
DateValue	Function
Day	Function
Format, Format\$	Function
Format	Property
GetOption, SetOption	Methods
KeyboardLanguage	Property
LabelAlign, TextAlign	Properties
LabelX, LabelY	Properties
Month	Function
NumeralShapes	Property
Orientation	Property
ReadingOrder	Property
ScrollBarAlign	Property
TransferText	Action/Method
Weekday	Function
Year	Function

For development of sophisticated right-to-left applications, you might want to take advantage of the numerous Microsoft Windows operating system

application programming interfaces (APIs) that have right-to-left settings. For more information, you can go to the Microsoft Office Web site.

[Show All](#)

About Arabic sort order

Arabic sort order is described here by using the Arabic code page, not [Unicode](#), because the former is more familiar to most users. The Arabic sort order table arranges the characters of the Arabic code page into a sort order. The sort order controls how data is sequentially ordered in Access databases that use the Arabic code page.

Note Access sorts right-to-left databases by using values from Unicode, not those from the right-to-left code page as in earlier versions. Opened or imported databases that use the Arabic code page and Arabic sort order will be converted to Unicode and [General sort order](#).

The 256 characters of the Arabic code page can be organized into several categories that determine how characters are sorted.

- [Kashida](#) and [bidirectional control characters](#) (hexadecimal code points 9D, 9E, DC, FD, and FE):

Sort ignores these characters.

- Arabic [diacritics](#) (hexadecimal code points F0 through F3, F5, F6, F8, and FA):

Sort ignores Arabic diacritics except when the strings being compared are otherwise identical. In this case, unmarked characters (without diacritics) sort ahead of marked characters (with diacritics); diacritics with lower code point values sort ahead of others.

- General control characters (hexadecimal code points 00 through 1F and 7F), reserved characters (hexadecimal code points 80, 81, 8D, 8E, 8F, and 90), and punctuation:

These three categories sort ahead of alphanumeric characters in category order (that is, General control characters first) and in ascending code point order within each category (for example, 00 sorts ahead of 01).

- Arabic and [left-to-right](#) alphanumeric characters (shown in the Arabic sort order table that follows):

Left-to-right text is always sorted ahead of Arabic text.

Left-to-right text sorting is not case sensitive. For example, although uppercase "A" is listed before lowercase "a" in the sort table, Access treats these characters equally and does not provide any sorting preference to either character.

Unmarked left-to-right text (without diacritics) is sorted before marked left-to-right text.

The following table lists the alphanumeric characters of the Arabic code page along with their respective hexadecimal code points. Characters (Char) are arranged in columns according to their sort order, starting in the upper-left corner.

Char	Code Point										
0	30	d	64	L	4C	T	54	ؤ	C4	ص	D5
1	31	E	45	l	6C	t	74	ا	C5	ض	D6
2	32	e	65	M	4D	U	55	ئ	C6	ط	D8
3	33	è	88	m	6D	u	75	آ	C2	ظ	D9
4	34	é	89	N	4E	ù	89	ا	C7	ع	DA
5	35	è	8A	n	6E	û	8B	ب	C8	ع	DB
6	36	ë	8B	O	4F	ü	8C	آ	B1	ف	DD
7	37	F	46	o	6F	V	56	ة	C9	ق	DE
8	38	f	66	Œ	8C	v	76	ت	CA	ك	DF
9	39	G	47	œ	9C	W	57	ث	CB	ك	90
A	41	g	67	ô	84	w	77	ج	CC	ل	E1
a	61	H	48	P	50	X	58	ح	8D	م	E3
À	C0	h	68	p	70	x	78	ح	CD	ن	E4
à	E0	I	49	Q	51	Y	59	ح	CE	ه	E5
â	E2	i	69	q	71	y	79	د	CF	و	E6
B	42	î	8E	R	52	ÿ	9F	ذ	D0	ى	EC
b	62	ï	8F	r	72	ÿ	FF	ر	D1	ي	ED
C	43	J	4A	S	53	Z	5A	ز	D2		
c	63	j	6A	s	73	z	7A	ل	8E		
ç	E7	K	4B	Š	8A	š	C1	س	DB		
D	44	k	6B	š	9A	š	C3	ش	D4		

[Show All](#)

About the Arabic code page

The feature described in this Help topic is only available if support for [right-to-left](#) languages is [enabled](#) through Microsoft Office Language Settings.

In addition, to enable the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, Microsoft Windows 2000.

Each [code page](#) is identified by a unique code page number. The code page (CP) for Arabic is CP 1256.

Access sorts right-to-left databases by using values from [Unicode](#), not those from the right-to-left code page as in earlier versions. Opened or imported databases that use the Arabic code page and Arabic sort order will be converted to Unicode and [General sort order](#).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		SP0	@	P	`	p	■	ع	SP	°	À	ذ	à	°		
1		!	1	A	Q	a	q	ق	،	±	ء	ر	ل	°		
2		"	2	B	R	b	r	,	'	¢	ز	آ	ز	â		
3		#	3	C	S	c	s	f	"	£	أ	س	م	'		
4		\$	4	D	T	d	t	"	"	¤	ؤ	ش	ن	ô		
5		§	5	E	U	e	u	...	•	¥	ح	ص	ه	°		
6		&	6	F	V	f	v	†	-	¦	ئ	ض	و			
7		´	7	G	W	g	w	‡	-	§	ا	x	ç	÷		
8		(8	H	X	h	x	^	~	¨	ب	ط	è	"		
9)	9	I	Y	i	y	%	™	©	ة	ظ	é	ù		
A		*	:	J	Z	j	z	Š	š	ª	ع	ع	ê	°		
B		+	;	K	[k	{	<	>	«	»	ث	غ	ë	û	
C		,	<	L	\	l		œ	œ	¬	¼	ح	-	ü		
D		-	=	M]	m	}	€		-	½	ح	ف	ı	ƒ	
E		.	>	N	^	n	~	ž	ž	®	¾	خ	ق	î	ł	
F		/	?	o	_	o	■	ÿ	¯	¿	د	ك	i	ÿ		

Legend  Code point contains a non-printable control character.  Code point is unoccupied; reserved for future use.

[Show All](#)

Specify the appearance of numbers in a text box or other control

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **NumeralShapes** box, do one of the following:
 - For numerals formatted according to [Unicode](#) rules for adjacent text, click **Context**. In general, this means numeral shapes are [Hindi](#) if surrounding text is Arabic.
 - For numerals formatted according to the **Numbers** tab in regional settings in Windows Control Panel, click **System**.
 - For Western (Arabic) numeral shapes, click **Arabic**.
 - For locale-specific numeral shapes, click **National**.

[Show All](#)

Set language-specific spelling options

1. On the **Tools** menu, click **Options**, and then click the **Spelling** tab.
2. If the dictionary language is not same as the language whose options you want to set, change to the appropriate language.

[How?](#)

1. On the **Tools** menu, click **Options**, and then click the **Spelling** tab.
2. Under **Dictionary Language**, select the language that you want to use when checking spelling.
3. Under **Language-specific**, do one of the following:
 - To turn on or off new German spelling rules, select or clear the **Use German post-reform rules** check box.
 - To specify Korean spelling options, select or clear the **Combine aux verb/adj.**, **Use auto-change list**, and **Process compound nouns** check boxes.
 - To specify Arabic spelling options, select the option you want in the **Arabic modes** box.

[Arabic modes](#)

Strict initial alef hamza ignores spelling rules regarding words beginning with an [alef hamza](#). When selected, the spelling checker will not distinguish between alefs with or without hamzas.

Strict final yaa ignores spelling rules regarding words ending with the letter yaa. When selected, the spelling checker will not distinguish between yaas with or without subscripted dots.

- To specify Hebrew spelling options, select the option you want in the **Hebrew modes** box.

[Hebrew modes](#)

Full designates the conventional script required by the Hebrew Language

Academy for writing text without diacritics.

Partial designates the traditional script that is currently used only for text with diacritics.

Mixed accepts any word, whether in Full or Partial script, or any nonconventional spelling variation. This mode highlights as potential mistakes only completely unrecognized words and typing errors.

Mixed authorized accepts words in both Full and Partial script, but highlights as potential mistakes any variation not permitted by either.

[Show All](#)

About Hebrew sort order

Note Access sorts right-to-left databases by using values from [Unicode](#), not those from the right-to-left code page as in earlier versions. Opened or imported databases that use the Hebrew code page and Hebrew sort order will be converted to Unicode and [General sort order](#).

Hebrew sort order is described here by using the Hebrew [code page](#), not Unicode, because the former is more familiar to most users. The Hebrew sort order table arranges the characters of the Hebrew code page into a sort order. The sort order controls how data is sequentially ordered in Access databases that use the Hebrew code page.

The 256 characters of the Hebrew code page can be organized into several categories that determine how characters are sorted.

- [Bidirectional control characters](#) (hexadecimal code points FD and FE):

Sort ignores these characters.

- Hebrew [diacritics](#) (hexadecimal code points C0 through CD, D1, and D2):

Sort ignores Hebrew diacritics except when the strings being compared are otherwise identical. In this case, unmarked characters (without diacritics) sort ahead of marked characters (with diacritics); diacritics with lower code point values sort ahead of others.

- General control characters (hexadecimal code points 00 through 1F and 7F), reserved characters (hexadecimal code points 80, 81, 8A, 8C through 90, 9A, 9C through 9F, CE through D0, D3 through DF, FB, FC, and FF), and punctuation:

These three categories sort ahead of alphanumeric characters in category order (that is, General control characters first) and in ascending code point order within each category (for example, 00 sorts ahead of 01).

- Hebrew and left-to-right alphanumeric characters (shown in the Hebrew

sort order table that follows):

Left-to-right text is always sorted ahead of Hebrew text.

Left-to-right text sorting is not case sensitive. For example, although uppercase "A" is listed before lowercase "a" in the table on the following page, Access treats these characters equally and does not provide any sorting preference to either character.

Unmarked left-to-right text (without diacritics) is sorted before marked left-to-right text.

The following table lists the alphanumeric characters of the Hebrew code page along with their respective hexadecimal code points. Characters (Char) are arranged in columns according to their sort order, starting in the upper-left corner.

Char	Code Point								
0	30	E	45	N	4E	W	57	ך	EA
1	31	e	65	n	6E	w	77	כ	EB
2	32	F	46	O	4F	X	58	ץ	EC
3	33	f	66	o	6F	x	78	ק	ED
4	34	G	47	P	50	Y	59	ת	EE
5	35	g	67	p	70	y	79	י	EF
6	36	H	48	Q	51	Z	5A	י	FO
7	37	h	68	q	71	z	7A	ו	F1
8	38	I	49	R	52	⌘	E0	ד	F2
9	39	i	69	r	72	י	E1	ק	F3
A	41	J	4A	S	53	י	E2	ג	F4
a	61	j	6A	s	73	י	E3	פ	F5
B	42	K	4B	T	54	י	E4	צ	F6
b	62	k	6B	t	74	י	E5	פ	F7
C	43	L	4C	U	55	י	E6	ך	F8
c	63	l	6C	u	75	י	E7	ו	F9
D	44	M	4D	V	56	י	E8	י	FA
d	64	m	6D	v	76	י	E9		

[Show All](#)

About the Hebrew code page

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support—for example, the Hebrew version of Microsoft Windows 2000.

Each code page is identified by a unique code page number. The code page (CP) for Hebrew is CP 1255.

Access sorts right-to-left databases using values from [Unicode](#), not those from the right-to-left code page as in earlier versions. Opened or imported databases that use the Hebrew code page and Hebrew sort order will be converted to Unicode and [General sort order](#).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		SP	0	Ⓜ	P	˘	p	■	■	SP	°	.	■	Ⓜ	ı	
1		!	1	A	Q	a	q	■	'	■	±	ı	ı
2		"	2	B	R	b	r	,	'	¢	²	ı	ı
3		#	3	C	S	c	s	f	"	£	³	...	■	ı	ı	ı
4		\$	4	D	T	d	t	"	"	¤	⁴	...	■	ı	ı	ı
5		%	5	E	U	e	u	...	•	¥	µ	...	■	ı	ı	ı
6		&	6	F	V	f	v	†	-	¦	¶	...	■	ı	ı	ı
7		'	7	G	W	g	w	‡	—	§	.	.	■	ı	ı	ı
8		(8	H	X	h	x	^	~	"	.	.	■	ı	ı	ı
9)	9	I	Y	i	y	‰	™	©	±	.	■	ı	ı	ı
A		*	:	J	Z	j	z	■	■	•	°	■	■	ı	ı	ı
B		+	;	K	[k	{	<	>	«	»	...	■	ı	ı	ı
C		,	<	L	\	l		■	■	-	¼	.	■	ı	ı	ı
D		.	=	M]	m	}	■	■	.	½	.	■	ı	ı	ı
E		.	>	N	^	n	~	■	■	©	¾	.	■	ı	ı	ı
F		/	?	O	_	o		■	■	-	■	■	■	ı	ı	ı

Legend

-  Code point contains a non-printable control character.
-  Code point is unoccupied; reserved for future use.

[Show All](#)

About using different calendars

Some of the content in this topic may not be applicable to some languages.

Access supports use of the Hijri calendar in addition to the standard Gregorian calendar in all locales except Thai and East Asian locales.

If you open an existing database with date references and you have selected the wrong calendar setting, the program might report a variety of errors or other unexpected behaviors. For example, if you have checked the **Hijri Calendar** setting, Access will display a syntax error message if a report makes reference to [Gregorian](#) dates that are invalid as [Hijri](#) dates (for example, the date 12/31/99 will produce a syntax error because no Hijri month has 31 days).

To avoid problems of this sort, all date references in a database must be based on the same calendar.

Important It is recommended that you choose the database calendar when you first create your database, and then never change it.

Other considerations when the database calendar is set to Hijri

- To ensure Hijri dates strings are evaluated correctly in expressions and Visual Basic for Applications statements, use the **CDate** function to evaluate the strings instead of declaring the strings as constants.

For example, type **CDate("1/25/20")** instead of **#1/25/20#**.

The **CDate** function will always return a value based on the active database calendar, whereas date constants are always evaluated by using the Gregorian calendar.

- For reliable behavior, enter and display dates in an unambiguous format. For example, dates in short date format might be misinterpreted in queries if the year or the day of the month are 12 or less (for example, 3/11/10). Dates in medium date format display only the first few characters of the month name, which can create ambiguity or an undesirable appearance. Use long date format whenever possible. You can alter the way a date is displayed in long format (for example, excluding the day of the month) by using the regional settings in Microsoft Windows **Control Panel**.
- Dates entered in the **ValidationRule** property of a Date/Time field in a table (in table [Design view](#)) must be Hijri. However, to have the **ValidationRule** date appear as a Hijri date in validation messages, the validation rule must use the **CDate** function (for example, 'CDate ("1/25/1420")').
- Sample Access applications, such as the Northwind database, are not localized and expect the database calendar to be Gregorian. Because the sample applications contain hard-coded date constants that are not valid Hijri dates, running the sample applications with the database calendar set to Hijri is not recommended.
- All Hijri years will be displayed as four-digit year values (for example, 1420) unless the **Format()** function is explicitly used.
- The Visual Basic **Date** function, used to set the system date for your computer, will not accept Hijri dates. If the database calendar is set to Hijri, set the calendar to Gregorian before setting system date with the **Date** function.

- Dates in tables exported as text files are always stored as Gregorian values. If the database calendar is set to Hijri, automatic Hijri to Gregorian conversion is performed during the export process. Similarly, dates in tables imported from text files are assumed to be Gregorian dates and will be converted to Hijri values during the import process if the database calendar setting is Hijri.

[Show All](#)

Switch database calendars

Any version of Access that supports right-to-left text (except Hebrew) also supports the use of the [Gregorian](#) (default) and [Hijri](#) calendar for date entry and display.

1. Click **Options** on the **Tools** menu, and then click the **International** tab.
2. Select the **Use Hijri Calendar** check box.

The setting is saved with the database and used for all date references in the database.

[Show All](#)

Summary of right-to-left properties for controls

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Most intrinsic Access controls can be configured to present right-to-left appearance and behavior. In some cases, this can be done by using standard Access control properties (for example, the **TextAlign** property) and repositioning of controls to achieve a right-to-left appearance.

However, in some cases right-to-left properties are needed to achieve proper right-to-left appearance and behavior. The following tables lists those controls for which new right-to-left properties are added:

	Label	Text box	Toggle button	Combo box	List box	Command button
TextAlign	X	X		X	X	
ReadingOrder	X	X	X	X	X	X
KeyboardLanguage		X		X		
ScrollBarAlign		X		X	X	
NumeralShapes	X	X		X	X	

[Show All](#)

Set right-to-left properties in controls

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

[Specify text reading order for a text box or other control](#)

You can select the type of [reading order](#) you want in a label, text box, toggle button, combo box, list box, or command button.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **ReadingOrder** box, do one of the following:
 - To have the reading order follow the language of the text first entered, click **Context**.
 - To force the reading order in one [direction](#), click **Right-To-Left** or **Left-To-Right**.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to dynamically change reading order and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is excluded because changing alignment might negatively affect appearance). When the focus is moved outside the control, the display format returns to the previous control reading order.

[Specify the initial keyboard language in a text box or combo box](#)

You can select the type of [keyboard language](#) you want in a text box or combo box.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **KeyboardLanguage** box, do one of the following:
 - To make the keyboard language follow the most recent keyboard language you used, click **System**.
 - To make the keyboard language a particular language, click one of the languages shown in the list.

Note The languages that appear in the **KeyboardLanguage** property box are determined by the keyboard languages you have installed. Available languages can be reviewed by double-clicking **Keyboard** or **Regional Options** in Microsoft Windows **Control Panel**.

[Specify text alignment in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **TextAlign** box, click one of the following:
 - To have text alignment follow the language used when the text was first entered or the [user interface language](#), click **General**.
 - To set a fixed alignment, click one of the other choices.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to dynamically change [reading order](#) and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is not changed because changing alignment might negatively affect appearance). When the focus leaves the control, the display format returns to the previous control reading order.

[Specify where you want scroll bars in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.

3. On the **All** tab, in the **ScrollBarAlign** box, do one of the following:
 - To have the scroll bar follow the setting of the **Orientation** property for the form or report that contains the control, click **System**. (A vertical scroll bar is placed on the left if the form or report **Orientation** property setting is **Right-to-left** and on the right if the form or report **Orientation** property setting is **Left-to-right**.)
 - To have the scroll bar on the right side of the control, click **Right**.
 - To have the scroll bar on the left side of the control, click **Left**.

 [Specify the appearance of numbers in a text box or other control](#)

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **NumeralShapes** box, do one of the following:
 - For numerals formatted according to [Unicode](#) rules for adjacent text, click **Context**. In general, this means numeral shapes are [Hindi](#) if surrounding text is Arabic.
 - For numerals formatted according to the **Numbers** tab in regional settings in Windows Control Panel, click **System**.
 - For Western (Arabic) numeral shapes, click **Arabic**.
 - For locale-specific numeral shapes, click **National**.

 [See a summary of right-to-left properties for controls](#)

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access 2000 (and later versions), you must be running a 32-bit Microsoft Windows operating system that has right-to-left support—for example, the Arabic version of Microsoft Windows 2000.

Most intrinsic Access controls can be configured to present right-to-left appearance and behavior. In some cases, this can be done by using standard Access control properties (for example, the **TextAlign** property) and repositioning of controls to achieve a right-to-left appearance.

However, in some cases right-to-left properties are needed to achieve proper right-to-left appearance and behavior. The following tables lists those controls for which new right-to-left properties are added:

	Label	Text box	Toggle button	Combo box	List box	Command button
TextAlign	X	X		X	X	
ReadingOrder	X	X	X	X	X	X
KeyboardLanguage		X		X		
ScrollBarAlign		X		X	X	
NumeralShapes	X	X		X	X	

[Show All](#)

Specify alignment behavior in text fields

1. On the **Tools** menu, click **Options** , and then click the **International** tab.
2. In the **General Alignment** option group, do one of the following :
 - Click **Interface Mode** to align text and numbers in text fields according to the [user interface language](#) in effect.
 - Click **Text Mode** to align text and numbers in text fields according to the [keyboard language](#) of the text first entered.

[Show All](#)

Specify the direction of a control on a data access page

1. From the [Database window](#), open the data access page in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Page Design** toolbar.
3. On the **All** tab, find the right-to-left property associated with the control (such as **Dir**).
4. Do one of the following:
 - Click **Right-to-left** (or the equivalent setting such as "rtl") to display the control with a right-to-left [direction](#).
 - Click **Left-to-right** (or the equivalent setting such as "ltr") to display the control with a [left-to-right](#) direction.

[Show All](#)

Specify text alignment in a text box or other control

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **TextAlign** box, click one of the following:
 - To have text alignment follow the language used when the text was first entered or the [user interface language](#), click **General**.
 - To set a fixed alignment, click one of the other choices.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to dynamically change [reading order](#) and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is not changed because changing alignment might negatively affect appearance). When the focus leaves the control, the display format returns to the previous control reading order.

[Show All](#)

Specify text reading order for a text box or other control

You can select the type of [reading order](#) you want in a label, text box, toggle button, combo box, list box, or command button.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **ReadingOrder** box, do one of the following:
 - To have the reading order follow the language of the text first entered, click **Context**.
 - To force the reading order in one [direction](#), click **Right-To-Left** or **Left-To-Right**.

Note CTRL+LEFT SHIFT and CTRL+RIGHT SHIFT shortcut keys used to dynamically change reading order and alignment in other [bidirectional](#) programs will temporarily change reading order only (alignment is excluded because changing alignment might negatively affect appearance). When the focus is moved outside the control, the display format returns to the previous control reading order.

[Show All](#)

Specify where you want scroll bars in a text box or other control

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **ScrollBarAlign** box, do one of the following:
 - To have the scroll bar follow the setting of the **Orientation** property for the form or report that contains the control, click **System**. (A vertical scroll bar is placed on the left if the form or report **Orientation** property setting is **Right-to-left** and on the right if the form or report **Orientation** property setting is **Left-to-right**.)
 - To have the scroll bar on the right side of the control, click **Right**.
 - To have the scroll bar on the left side of the control, click **Left**.

[Show All](#)

About attaching labels to controls

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Labels are automatically created and attached to most controls. Label placement is determined by the **Orientation** property setting for the parent form at the time you create the label. If the **Orientation** property is set to right-to-left, the label is placed to the right of the control. If the **Orientation** property is set to [left-to-right](#), the label is placed to the left of the control.

Properties for a label attached to a control, except **TextAlign**, are inherited from default label control properties. **TextAlign** for an attached label is inherited from the **LabelAlign** property in the parent control (for example, text box, option group, toggle button, option button, check box, combo box, list box, subform/subreport, bound object frame, or command button). **LabelAlign**, which specifies the alignment of text within a label, exists only as a parent property until the control is created. Thereafter, it is available as **TextAlign** in the label control attached to the parent control.

[Show All](#)

Specify the initial keyboard language in a text box or combo box

You can select the type of [keyboard language](#) you want in a text box or combo box.

1. Open a form or report in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Design** toolbar.
3. On the **All** tab, in the **KeyboardLanguage** box, do one of the following:
 - To make the keyboard language follow the most recent keyboard language you used, click **System**.
 - To make the keyboard language a particular language, click one of the languages shown in the list.

Note The languages that appear in the **KeyboardLanguage** property box are determined by the keyboard languages you have installed. Available languages can be reviewed by double-clicking **Keyboard** or **Regional Options** in Microsoft Windows **Control Panel**.

[Show All](#)

About converting reports saved as text files from ANSI characters

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

Access reports exported as text files using the **Export** command (**File** menu) are saved as simple text files with characters encoded as ANSI code page values.

Viewing reports that contain bi-directional text and are saved as text files

Reports exported as text files using the **Export** command (**File** menu) are saved as simple text files with fields in logical, not visual order. You can use the resulting file with a number of different programs, but the file can be hard to read when it contains right to left text. Depending on the report complexity, a viewable version of the report can frequently be obtained by opening the file in Notepad and switching document reading order by using CTRL+LEFT-SHIFT and CTRL+RIGHT-SHIFT.

[Show All](#)

Convert bidirectional text files from ANSI

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

The ConvText utility provides more support (additional code pages and [visual](#) text layout options) than is found in some Microsoft Office applications to convert [bidirectional](#) files from [ANSI](#) characters. ConvText is available on the Microsoft Office Web site and on the installation CDs for Microsoft Office language versions that offer right-to-left support.

The following example illustrates how to use ConvText with Access. Similar steps are required to use ConvText with other Microsoft Office applications such as Excel.

1. Export the Access object you want to convert to a non-ANSI code page.
2. In the Program Files\Microsoft Office\Office folder, double-click the **ConvText.exe** icon.
3. Click **Convert From Windows ANSI Code Page**.
4. Type file names in the **Source** and **Target** boxes, or click **Browse**.
5. Do one of the following:
 - Click **Load** to retrieve a previously saved conversion specification.
 - Select the code page, text layout, file type, and any other relevant parameters that match the target text file to be created.
6. Click **Save** to retain the conversion specifications.
7. Click **Convert**.

[Show All](#)

Convert bidirectional text files to ANSI

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

The ConvText utility provides more support (additional code pages and [visual](#) text layout options) than is found in some Microsoft Office applications to convert [bidirectional](#) files to [ANSI](#) characters. ConvText is available on the Microsoft Office Web site and on the installation CDs for Microsoft Office language versions that offer right-to-left support.

The following example illustrates how to use ConvText with Access. Similar steps are required to use ConvText with other Microsoft Office applications such as Excel.

1. In the Program Files\Microsoft Office\Office folder, double-click the **ConvText.exe** icon.
2. Click **Convert To Windows ANSI Code Page**.
3. Type file names in the **Source** and **Target** boxes, or click **Browse**.
4. Do one of the following:
 - Click **Load** to retrieve a previously saved conversion specification.
 - Select the code page, text layout, file type, and any other relevant parameters that match the target text file to be created.
5. Click **Save** to retain the conversion specifications.
6. Click **Convert**.
7. Import the converted text file into Access.

All field names and records from the original table or worksheet will be fully recreated in a new table.

[Show All](#)

About right-to-left features in data access pages

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Right-to-left features in data access pages are set according to a two-level hierarchy: page level and control level.

[Page level](#)

Initially, a new data access page will be displayed in the [direction](#), either [left-to-right](#) or right-to-left, that is selected under **Default direction** on the **International** tab in the **Options** dialog box (**Tools** menu). That direction appears as a setting of the **Dir** property of the data access page.

Note Data access page direction is included in the underlying HTML code of the data access page through use of the standard HTML Direction (DIR) attribute. For example, the code `<HTML DIR=RTL>` placed at the beginning of the `<HTML>` element identifies a right-to-left data access page.

Controls placed on a data access page inherit the directionality of the data access page itself, depending on whether the control has right-to-left features. Controls without right-to-left features are displayed in a left-to-right direction.

[Control level](#)

Controls with right-to-left features initially appear on a data access page according to the direction of the data access page. After you place a control on a data access page, you might be able to change the control display direction to give it right-to-left appearance and functionality. The property name that controls

display direction varies with control type. The properties in the following table control directionality for data access page controls:

Control type	Description
	The Office Web Component controls are the spreadsheet, the PivotChart, and the PivotTable.
Office Web Component controls	The direction setting property is RightToLeft . Note Office Web Component controls each have a property toolbox that can be used both while designing the control and in the browser; however, the right-to-left setting can't be changed in the browser.
Intrinsic browser controls	Examples of intrinsic browser controls include Table, Label, TextBox, ComboBox, ListBox, OptionGroup, and ScrollingText. The direction setting property is typically Dir .
ActiveX controls	These are other controls that are available with Microsoft Office programs or Microsoft Visual Basic. The direction setting property is typically RightToLeft .

[Show All](#)

Use right-to-left features in data access pages

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.



[Set the direction for new tables, queries, forms, reports, and data access pages](#)

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. In the **Default Direction** option group, do one of the following:
 - Click **Left-To-Right** to have new objects displayed in a [direction](#) familiar to [left-to-right](#) users.
 - Click **Right-To-Left** to have new objects displayed in a direction familiar to right-to-left users.



[Specify the direction of a data access page](#)

Do one of the following:

- For a new [data access page](#):

Before creating a new data access page, set the [direction](#), either [left-to-right](#) or right-to-left, under **Default direction** on the **International** tab in the **Options** dialog box (**Tools** menu). That direction will appear as the setting of the **Dir** property of the new data access page.

- For an existing data access page:

1. From the [Database window](#), open the data access page in [Design view](#).

2. Select the page by clicking the page title bar.
3. On the **Page Design** toolbar, click **Properties** , click the **All** tab, and change the **Dir** property setting to the page direction you want: **rtl** for right-to-left or **ltr** for left-to-right.

[Tip](#)

To set text direction directly in HTML, click **HTML Source** on the **View** menu and modify the <HTML> element to <HTML DIR=RTL> for a right-to-left page direction or <HTML DIR=LTR> for a left-to-right page direction.

[Specify the direction of a control on a data access page](#)

1. From the [Database window](#), open the data access page in [Design view](#).
2. Make sure the control is selected, and then click **Properties**  on the **Page Design** toolbar.
3. On the **All** tab, find the right-to-left property associated with the control (such as **Dir**).
4. Do one of the following:
 - Click **Right-to-left** (or the equivalent setting such as "rtl") to display the control with a right-to-left [direction](#).
 - Click **Left-to-right** (or the equivalent setting such as "ltr") to display the control with a [left-to-right](#) direction.

[Show All](#)

Specify the direction of a data access page and its controls

Do one of the following:

- For a new [data access page](#):

Before creating a new data access page, set the [direction](#), either [left-to-right](#) or right-to-left, under **Default direction** on the **International** tab in the **Options** dialog box (**Tools** menu). That direction will appear as the setting of the **Dir** property of the new data access page.

- For an existing data access page:
 1. From the [Database window](#), open the data access page in [Design view](#).
 2. Select the page by clicking the page title bar.
 3. On the **Page Design** toolbar, click **Properties** , click the **All** tab, and change the **Dir** property setting to the page [direction](#) you want: **rtl** for right-to-left or **ltr** for left-to-right.



To set text direction directly in HTML, click **HTML Source** on the **View** menu and modify the <HTML> element to <HTML DIR=RTL> for a right-to-left page direction or <HTML DIR=LTR> for a left-to-right page direction.

[Specify the direction of a control on a data access page](#)

[Show All](#)

About right-to-left behavior in queries

Right-to-left characteristics for queries are summarized as follows:

[In Query Design view](#)

[Query Design view](#) displays the query in a layout consistent with the current [user interface language](#).

All Text and Memo fields are assigned General alignment and Context [reading order](#). Visual layout adjusts automatically based on the language of the first character entered in a field. Alignment can be overridden by creating a custom format by using the format control character, the exclamation point (!), in the **Format** property in the query's property sheet. Reading order, however, is fixed.

Access provides right-to-left support for many of the other query features available in Design view. For example, if you choose to design a crosstab query, you have the same ability to use the query's property sheet to change the view direction of the crosstab query you're creating.

[In Query Datasheet view](#)

The direction of the data, description fields, and other window components will be right-to-left or left-to-right depending on the query orientation you choose in Design view.

The **Orientation** property allows you to change the overall right-to-left or left-to-right [direction](#) of the query in Datasheet view, regardless of the **Default direction** setting on the **International** tab in the **Options** dialog box (**Tools** menu).

[Other query design considerations](#)

You can use right-to-left or mixed right-to-left and left-to-right text to name

many of the objects you create. You can:

- Provide [bidirectional](#) text in names given to tables you create with a make-table query.
- Select tables with right-to-left or bidirectional text names when you add data to a table by using an append query.
- Name query parameters with right-to-left or bidirectional text in a parameter query.



The editor displays SQL statements in an overall left-to-right direction regardless of the user interface language or chosen query view direction.

Important Dates in SQL statements are always displayed as [Gregorian](#) calendar values, regardless of database calendar chosen. If you change a date in a SQL statement, the new value must be a Gregorian date.

[Show All](#)

Change query direction

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

[Change individual query direction](#)

[To change direction in Design view](#)

1. Open a query in [Design view](#).
2. Click anywhere in query Design view outside the design grid and the field lists.
3. On the **Query Design** toolbar, click **Properties** .
4. In the **Orientation** property box, do one of the following:
 - For a right-to-left direction, click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-right**.

[To change direction in Datasheet view](#)

1. Open a query in [Datasheet view](#).
2. Click anywhere in the text field.
3. On the **Format** menu, click **Datasheet**.
4. In the **Direction** option box, do one of the following:
 - For a right-to-left direction, click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-right**.

 [Change the default direction for new queries](#)

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. In the **Default Direction** option group, do one of the following:
 - Click **Left-To-Right** to have new objects displayed in a [direction](#) familiar to [left-to-right](#) users.
 - Click **Right-To-Left** to have new objects displayed in a direction familiar to right-to-left users.

[Show All](#)

Change individual query direction

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

[To change direction in Design view](#)

1. Open a query in [Design view](#).
2. Click anywhere in query Design view outside the design grid and the field lists.
3. On the **Query Design** toolbar, click **Properties** .
4. In the **Orientation** property box, do one of the following:
 - For a right-to-left direction, click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-right**.

[To change direction in Datasheet view](#)

1. Open a query in Datasheet view
2. Click anywhere in the text field.
3. On the **Format** menu, click **Datasheet**.
4. In the **Direction** option box, do one of the following:
 - For a right-to-left direction, click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-right**.

[Show All](#)

About right-to-left behavior in tables

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

Right-to-left characteristics for tables are summarized as follows:

[New table direction](#)

The direction for new tables is determined by the **Default direction** setting on the **International** tab in the **Options** dialog box (**Tools** menu).

[Existing table direction](#)

For tables in a [Microsoft Access database](#) (.mdb), the direction of an existing table can be "flipped" from right to left, or the reverse, in [Design view](#) by changing the table's **Orientation** property. In Datasheet view, table direction in both an Access database and a [Microsoft Access project](#) (.adp) can be "flipped" by setting **Direction** in the Datasheet dialog box (**Format** menu).

[User interface direction](#)

The direction and layout of several features that support tables, such as the [Design view](#) window, the Index window, frozen columns, and key visual characteristics such as Field Properties descriptions and tabs, the current record indicator, and grid column order are determined by the [user interface](#) you are currently using.

[Field properties](#)

Text and Memo fields are assigned General alignment and Context [reading order](#). Visual layout is automatically adjusted based on the language of the first

character entered in a field when General alignment is set to Text mode in the **International** tab in the **Options** dialog box (**Tools** menu). Alignment can then be overridden by creating a custom format by using the format control character, the exclamation point (!), in the **Format** property. Reading order, however, cannot be changed. You can create tables with more sophisticated right-to-left field properties and data entry features if you create a form that uses your table as a record source, and then view the form in [Datasheet view](#).

[Show All](#)

Change table direction

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support - for example, Microsoft Windows 2000.

[Change individual table direction](#)

[In table Design view](#)

1. Open a table in [Design view](#).
2. On the **Table Design** toolbar, click **Properties** .
3. Click the **Orientation** property.
4. Do one of the following:
 - For a right-to-left [direction](#), click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-Right**.

[In table Datasheet view](#)

1. Open a table in [Datasheet view](#).
2. Click anywhere in the text field.
3. On the **Format** menu, click **Datasheet**.
4. In the **Direction** option box, do one of the following:
 - For a right-to-left [direction](#), click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-Right**.

[Change the default direction for new tables](#)

1. On the **Tools** menu, click **Options**, and then click the **International** tab.
2. In the **Default Direction** option group, do one of the following:
 - Click **Left-To-Right** to have new objects displayed in a [direction](#) familiar to [left-to-right](#) users.
 - Click **Right-To-Left** to have new objects displayed in a direction familiar to right-to-left users.

[Show All](#)

Change individual table direction

The feature described in this Help topic is only available if support for [right-to-left](#) languages is enabled through Microsoft Office Language Settings.

In addition, to [enable](#) the right-to-left features in Microsoft Access, you must be running a 32-bit Microsoft Windows operating system that has right-to-left support— for example, the Arabic version of Microsoft Windows 2000.

[In table Design view](#)

1. Open a table in [Design view](#).
2. On the **Table Design** toolbar, click **Properties** .
3. Click the **Orientation** property.
4. Do one of the following:
 - For a right-to-left [direction](#), click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-Right**.

[In table Datasheet view](#)

1. Open a table in [Datasheet view](#).
2. Click anywhere in the text field.
3. On the **Format** menu, click **Datasheet**.
4. In the **Direction** option box, do one of the following:
 - For a right-to-left [direction](#), click **Right-to-Left**.
 - For a [left-to-right](#) direction, click **Left-to-Right**.



View information on object dependencies

[What's new home](#)

In Microsoft Office Access 2003, you can view information on dependencies between database objects. Viewing a list of objects that use a specific object helps maintain a database over time and avoid errors related to missing record sources. For example, the Quarterly Orders query in the Sales database is no longer needed, but before deleting it, you might want to find out which other objects in the database use the query. Then, you could either change the record source of the dependent objects, or delete them, before deleting the Quarterly Orders query. Viewing a complete list of dependent objects helps you save time and minimize errors.

In addition to viewing the list of objects that are bound to a selected object, you can also view the objects that are being used by the selected object.

Macros, modules, and data access pages are not searched for dependencies. Access projects do not support this feature.



Error checking in forms and reports

[What's new home](#)

In Microsoft Office Access 2003, you can enable automatic error checking for common errors in forms and reports. Error checking points out errors, such as two controls using the same keyboard shortcut, and the width of a report being greater than the page it will be printed on. Enabling error checking helps you identify errors and correct them.

The following illustration shows the BirthDate label control in Design view. Error checking for errors related to shortcut keys is turned on, so you see an error indicator on the top left corner of the control. The control has a duplicate shortcut key assigned to it.



If you select the control with an error indicator, the **Error Checking Options** button appears.



If you rest your mouse pointer on the **Error Checking Options** button and then click the arrow in the button, a menu displays a set of options to correct or ignore the error.



Propagating field properties

[What's new home](#)

In previous versions of Microsoft Access, whenever you modified a field's inherited property, you had to manually modify the property of the corresponding control in each of the forms and reports. Now, when you modify an inherited field property in Table design view, Access displays an option to update the property of all or some controls that are bound to the field.



Smart tags

[What's new home](#)

In Microsoft Office Access 2003, you can use the **SmartTags** property to add a smart tag to any field in a table, query, form, report, or data access page in a database.



Back up a database or project

[What's new home](#)

You can back up the current database or project before making major changes to it. The backup will be saved in the default backup location, or in the current folder.

To restore a database, go to the location of the backup, rename the file, and open it in Access.



Windows XP theme support

[What's new home](#)

The Microsoft Windows XP operating system offers you several themes. If you have chosen a theme other than the default, Access will apply the chosen theme to views, dialog boxes, and controls. You can prevent form controls from inheriting themes from the operating system by setting an option on the database or project.



Improved sorting in controls

[What's new home](#)

You can now specify the ascending or descending sort order of up to four fields in the List Box and Combo Box Wizards in forms and reports, and the Lookup Wizard in an Access database. The sort page added to these wizards looks and behaves like the sort page in the Report Wizard.



Autocorrect options

[What's new home](#)

In Microsoft Office Access 2003, you have more control over the behavior of the AutoCorrect feature. The **AutoCorrect Options** button appears near text that was automatically corrected. If you find on occasion that you don't want text to be corrected, you can undo a correction or turn **AutoCorrect** options on or off by clicking the button and making a selection.



Enhanced font capabilities in SQL views

[What's new home](#)

In the SQL and query Design views of a query in both a Microsoft Access database and Microsoft Access project, you can now change the font and font size of the text by using the **Query design font** option added to the **Tables/Queries** tab of the **Options** dialog box under the **Tools** menu. These settings apply to all databases and work with the high-contrast and other accessibility settings of your computer.



Context-based Help in SQL view

[What's new home](#)

In the SQL view of a query in a Microsoft Access database, you can now get help specific to Jet SQL keywords, VBA functions, and Access functions. Simply press F1 to bring up the help that corresponds to the text near the cursor. You can also search the Jet SQL and VBA function reference topics.



Importing, exporting, and linking

[What's new home](#)

Importing, exporting, and linking to a Microsoft Windows SharePoint Services list from Access

You can perform the following operations with a Windows SharePoint Services list:

- Export the contents of a table or a query to a list.
- Import the contents of a list into a table.
- Link a table to a list.

Exporting and linking to Access data from Windows SharePoint Services

You can now export a list in its Datasheet view from Windows SharePoint Services to a static table or to a linked table in Access. When you export to a static table, you create a table in Access. You can then view and make changes to the table independent of the original list in Windows SharePoint Services. Similarly, you can change the list in Windows SharePoint Services, and that will not affect the table in Access.

When you export to a linked table, you create a table in Access and establish a dynamic link between the table and the list such that changes to the table are reflected in the list, and changes to the list are reflected in the table as well.

Make a local table from a linked table

In Microsoft Office Access 2003, you can make a local copy of the structure or data and structure contained in a linked table.



XML support

[What's new home](#)

With the enhanced XML support in Microsoft Office Access 2003, you can specify a transform file when you import data from or export data to XML. The transform is then applied automatically. When you import XML data, the transform is applied to the data as soon as the data is imported, before any new table is created or an existing one is appended to. When you export data to XML, the transform is applied following the export operation.

Often times a database contains lookup values that are stored in another database. You can now include these related tables when exporting. You can also include any predefined filter or sort order for an object when exporting the object.

[Show All](#)



Security enhancements

[What's new home](#)

Macro Security Microsoft Office Access 2003 allows you to protect against potentially unsafe Visual Basic for Applications (VBA) code by setting the macro security level. You can set the security level so that you are prompted every time that you open a database containing VBA code, or you can automatically block databases that are from unknown sources.

Additionally, Access uses Microsoft Authenticode technology to enable you to [digitally sign](#) a [macro project](#) by using a [digital certificate](#). The certificate used to create this signature confirms that the macro originated from the signer, and the signature confirms that it has not been altered. When you set the macro security level, you can run macros based on whether they are digitally signed by a developer on your list of trusted sources.

Block Potentially Unsafe Functions Access utilizes the Microsoft Jet Expression Service enhanced sandbox mode to block potentially unsafe functions from being used in expressions.



More new features

[What's new home](#)

New look for Office Microsoft Office 2003 has a new look that's open and energetic. Additionally, new and improved task panes are available to you. New task panes include Getting Started, Help, Search Results, and Research.

Tablet PC support On a Tablet PC, you can quickly provide input using your own handwriting directly into Office documents as you would using a pen and a printout. Additionally, you can now view task panes horizontally to help you do your work on the Tablet PC the way you want to do your work.

Microsoft Office Online Microsoft Office Online is better integrated in all Microsoft Office programs so that you can take full advantage of what the site has to offer while you work. You can visit Microsoft Office Online directly from within your Web browser or use the links provided in various task panes and menus in your Office program to access articles, tips, clip art, templates, online training, downloads, and services to enhance how you work with Office programs. The site is updated regularly with new content based on direct feedback and specific requests from you and others who use Office.

Improving quality for the customer Microsoft strives to improve quality, reliability, and performance of Microsoft software and services. The Customer Experience Improvement Program allows Microsoft to collect information about your hardware configuration and how you use Microsoft Office programs and services to identify trends and usage patterns. Participation is optional, and data collection is completely anonymous. Additionally, error reporting and error messages have been improved so that you are provided with the easiest approach to reporting errors and the most helpful information about alerts at the time you encounter a problem. Finally, with an Internet connection, you can give Microsoft customer feedback about an Office program, help content, or Microsoft Office Online content. Microsoft is continually adding and improving content based on your feedback.

Add Table Dialog Box (Database Designer)

Lets you add tables in Database Designer.

Add

Adds the selected table or tables.

Note If you want to add several tables to the database diagram, you can select them all before clicking Add. Alternatively, you can double-click each table you want to add, then click Close when you are finished.

Close

Closes the dialog box without adding further tables.

Add Table Dialog Box (Query Designer)

Lets you add tables, views, or user-defined functions returning a table in the Query Designer.

Tables

Lists the tables you can add to the Diagram pane. To add a table, select it and click **Add**. To add several tables at once, select them and click **Add**.

Views

Lists the views you can add to the Diagram pane. To add a view, select it and click **Add**. To add several views at once, select them and click **Add**.

Functions

Lists the user-defined functions you can add to the Diagram pane. To add a function, select it and click **Add**. To add several functions at once, select them and click **Add**.

Add

Adds the selected item or items.

Close

Closes the dialog box without adding further items, including the currently highlighted items.

Choose Name Dialog Box

Lets you change the system-assigned name of a new table before that table is created in the database.

Enter a name for the table

Displays the system-assigned name of the table. Type a new name for the table in the text box. Although you can later use the Table property page to change the system-assigned name, it is best if you make any name changes now. Changing the name of an existing database object is risky, because you can invalidate programs and queries that refer to that object by name.

OK

Creates the table with the name you specified. The table is created in the database when you save the table or database diagram.

Cancel

Cancels the creation of the new table.

For more information about creating tables, see [Add Tables to a Database Diagram or Database](#).

See Also

[Rename a Table in a Database Diagram](#)

Choose Target Table for Insert Results Dialog Box

Allows you to specify a table to insert new rows into. This dialog box appears when you start an Append query. You can create an Append query only as part of a stored procedure.

Table name

Select from the list the name of the table to add rows to. You can specify only one table for the Append query.

Note You can change the table into which you want to insert rows in the Properties window. For details, see [Query Property Page](#).

Choose Target Table for Insert Values Dialog Box

Allows you to specify the table to add a row to. This dialog box appears if more than one table is displayed in the Diagram pane when you change a query's type to be an Append query. You can create an Append query only as part of a stored procedure.

Select the table to add a row to, and then choose **OK**.

Column Selection Dialog Box

Lets you change the Custom view for tables in the database diagram. Custom view shows only the column properties identified by the user.

This dialog box appears when you point to the **Diagram** menu, right-click a table and then click **Modify Custom View**.

Available columns

Lists all columns existing in the selected database table. The columns listed here depend on the properties of the database table and the type of database.

Highlight the desired column and use the arrow buttons to move the columns to the **Selected columns** box.

Selected columns

Displays the column properties currently defined for the Custom view. Use the arrow buttons to add and remove column properties to the Selected Columns list.

Move

Use the up and down arrow buttons to move highlighted columns up or down in the **Selected columns** list. The column properties will be displayed in the table in the order shown in the **Selected columns** list.

Save as default

Replaces the default Custom view with the columns selected in this dialog box. If not selected, the column selection specified in the dialog box will be applied only to the selected table in the database diagram.

OK

Saves the Custom view.

Cancel

Cancels the modification of Custom view.

For more information about Custom views, see [Change a Table View in a Database Diagram](#).

Create Relationship Dialog Box

Lets you confirm the related columns and to set properties for a new relationship.

This dialog box appears when you draw a relationship line between two tables in your database diagram.

Relationship name

Displays the system-assigned name of the relationship. To rename the relationship, type a new name in the text box.

Primary key table

Shows the name of the primary key table in the relationship, followed by the columns that make up the primary key. You can select different columns to match the columns shown under **Foreign key table**.

Foreign key table

Shows the name of the foreign key table in the relationship, followed by the columns that make up the foreign key. You can select different columns to match the columns shown under **Primary key table**.

Check existing data on creation

Applies the relationship to existing data in the foreign key table when the relationship is created. If you select this box, an error message will notify you if any data that violates the constraint exists.

Enforce relationship for replication

Selecting this option enforces referential integrity for the relationship whenever the foreign key table is replicated to a different database.

Enforce relationship for INSERTs and UPDATEs

Selecting this options enforces the constraint whenever data is added to or updated in the foreign key table using these statements.

- **Cascade update related fields (Microsoft SQL Server 2000 only)**

Instructs the database to propagate new key values to corresponding foreign key fields whenever a primary key value is updated.

- **Cascade delete related fields (SQL Server 2000 only)**

Instructs the database to delete corresponding rows from the foreign key table whenever rows from the primary key table are deleted.

Validation Text

Shows the text that is displayed to the users whenever a row that violates the foreign-key constraint is entered.

OK

Creates the relationship with the properties you selected.

Cancel

Erases the relationship line from your database diagram. The relationship is not created.

For more information about creating relationships between tables, see [Create a Relationship Between Tables](#).

Database Changes Detected Dialog Box (Database Designer)

Appears if you attempt to save a database diagram or selected tables but some of the database objects that will be affected by the save action are out of date with the database. Accepting the changes shown in this dialog box updates the database to match your diagram and overwrites other users' changes.

Note Although you cannot undo changes made to a table or database diagram, the changes are not saved to the database until you save the table or database diagram. You can discard any unsaved changes by choosing **No** and closing all open database diagrams without saving them.

Yes

Updates the database with all the changes shown in the list.

No

Cancels the save action.

Save Text File

Displays the **Save As** dialog box, letting you specify a location for a text file containing a list of the database changes.

For more information about saving database objects that have changed since you began working in the Database Designer, see [Reconcile a Database Diagram with a Modified Database](#) and [Interactions Between Database Diagrams, Table Design Windows, and the Database](#).

Datatype Change Required Dialog Box

Appears when you change the data type, length, scale, precision, or collation of a column that participates in a relationship.

Yes

Changes the data type property of the related columns shown in the list so that existing relationships are preserved.

No

Cancels the data type change and restores the previous data type to the column you just changed.

Define Column Collation Dialog Box

(Microsoft SQL Server 2000 only.) Lets you specify a collation sequence for the column. A column's collation sequence is used in any operation that compares values of the column to each other or to constant values. It also affects the behavior of some string functions, such as SUBSTRING, and CHARINDEX. For a complete list of the effects of a column's collation setting, see the SQL Server documentation.

This dialog box appears in any of several situations:

- If you enter an invalid collation name in the **Collation** box in the bottom portion of the Table Designer or in the **Columns** property page
- If you click in the **Collation** box in the Columns property page or in the bottom portion of the Table Designer, then click the button appearing to the right of the control.

SQL Collation

Select SQL Collation if you want to choose among the collation sequences defined by SQL Server. Then select a collation sequence from the drop-down list.

Windows Collation

Select Windows Collation if you want to choose among the collation sequences defined by Windows. Then select a collation sequence from the drop-down list.

Binary Sort

Applies only if you select Windows collation. Select this if you want comparison operations to use the binary codes of character values for comparison. If you select this option, certain alphabetic comparison options become unavailable. For example, making case-insensitive comparisons becomes unavailable because uppercase letters and lowercase letters have different binary encodings.

Dictionary Sort

Applies only if you select Windows collation. Select this if you want comparison operations to use certain alphabetic comparison options. The alphabetic comparisons options are:

- **Case Sensitive** Select this if you want comparisons to consider uppercase and lower case letters to be unequal.
- **Accent Sensitive** Select this if you want comparisons to consider accented and unaccented letters to be unequal. If you select this, comparisons will also consider differently accented letters to be unequal
- **Kana Sensitive** Select this if you want comparisons to consider katakana and hiragana Japanese syllables to be unequal.
- **Width Sensitive** Select this if you want comparisons to consider half-width and full-width characters to be unequal.

Restore Default

Applies to the column the default collation sequence for the database.

Delete Table Dialog Box

Allows you to specify the table from which to delete rows. This dialog box appears if more than one table is displayed in the Diagram pane when you change the type of a query to a Delete query. You can change the type of a query to a Delete Query only if that query defines a single-statement stored procedure.

Select the table to delete rows from, and then choose OK.

Note A Delete query removes entire rows from the table. If you want to clear values from individual data columns, use an Update query. If you want to delete columns from a table definition, or delete the table itself, use the Database Designer.

See Also

[Create a Delete Query](#) | [Create an Update Query](#)

Indexes Dialog Box

(Microsoft SQL Server 2000 only.) Allows you to create, delete, and modify indexes. This dialog box applies only to views.

Selected Index

Shows the name of the first index for the selected view. To show properties for a different index, select an index from the drop-down list.

New

Choose this button to create a new index. For more information, see [Create Indexed Views](#).

Delete

Choose this button to remove the selected index.

Index Name

Shows the name of the selected item. You can rename the item by entering a new name in this box.

Column Name/Order

Shows the columns contributing to the index, along with whether each column's values are arranged in ascending or descending order within the item. You can add, change, or remove column names in this list. You can also change the ascending/descending setting for each column.

Index File Group

Select the name of the file group in which you want to store the selected index. You must have at least one user-defined file group for this setting to be enabled. This setting is only available for Microsoft SQL Server 7.0 or higher databases. If you create a database object and do not specify its file group, SQL Server will assign it to the default file group. Initially, the default file group is the Primary

file group.

For more information on creating and using file groups, see your SQL Server documentation.

Create UNIQUE

Select this option to create a unique index for the selected view.

- **Ignore duplicate key** If you create a unique index, you can set this option to ensure each index value is unique.

Create as CLUSTERED

Select this option to create a clustered index for the selected view. For more information, see [Create a Clustered Index](#).

Do not automatically recompute statistics

Select this option to tell SQL Server to use previously created statistics. This choice, available only for SQL Server 7.0 and higher databases, may not produce optimal results and is not recommended.

For more information, see your SQL Server documentation.

Fill factor

Shows the fill factor that specifies how full each index page can be. If a fill factor is not specified, the database's default fill factor is used. For more information, see [Specify a Fill Factor for an Index](#).

Pad Index

If you specified a fill factor of more than zero percent, and you selected the option to create a unique index, you can tell SQL Server to use the same percentage you specified in fill factor as the space to leave open on each interior node. By default, SQL Server sets a two-row index size.

Make-Table Dialog Box

Allows you to name a table that will be created and that you will copy rows to. This dialog box appears when you change a query's type to be a Make-Table query. You can create a Make-Table query only as part of a stored procedure.

Table Name

Type the name of the table to create. The Query Designer does not check whether the name is already in use or whether you have permission to create the table.

To create a destination table in another database, specify a fully qualified table name, including the name of the target database, the owner (if required), and the name of the table.

Note Before you execute the query, from the Property Pages window, you can change properties of the table you want to create. For details, see [Query Property Page](#).

See Also

[Create a Make-Table Query](#)

Page Setup Dialog Box

Use this dialog box to specify a diagram's printer settings, such as orientation, scale, and paper size. The print setup options you set for a diagram are saved with the diagram file. When you change the Print scale and Orientation options for the diagram, the page breaks for the diagram are automatically updated.

Name

Shows the name of the default printer and lists the printers that are set up on your computer.

Properties

Sets the options for the selected printer. The options available depend on the features of the printer.

Status

Shows information about the selected printer.

Type

Displays the model and make information for the selected printer.

Where

Displays the network path or serial port information for the selected printer.

Comment

Displays additional information about the selected printer.

Size

Specifies the size of the paper or envelope you want to use.

Source

Specifies where the paper you want to use is located in the printer. Different printer models support different paper sources, such as the upper tray, envelope feed, and manual feed.

Portrait

Sets a taller-than-wide layout for the document on the page.

Landscape

Sets a wider-than-tall layout for the document on the page.

Percent

Sets the size of the diagram based on the settings as applied to a 100-percent diagram. Choose a percentage between 10 and 400.

Post-Save Notifications Dialog Box

Appears when errors are encountered while trying to save a database diagram or selected tables. This dialog box lists the following: the objects that were successfully saved in the database, the objects that were not saved, and the errors that were encountered.

OK

Returns to the database diagram.

Save Text File

Displays the **Save As** dialog box, prompting you for a location in which to save a text file listing the tables. This file provides a record of the database changes that were successfully saved as well as the changes that could not be saved due to errors.

See Also

[About Database Designer Considerations for Microsoft SQL Server Databases](#)

[Show All](#)

Query Definitions Differ Dialog Box

Notifies you that your query cannot be represented graphically in the Diagram and Grid panes and that you can edit your query only in the SQL pane. This dialog box appears when you enter or edit an SQL statement in the SQL pane; then you either switch to another pane, verify the query, or attempt to execute the query; and one of the following conditions applies:

- The SQL statement is incomplete or contains one or more syntax errors.
- The SQL statement is valid but is not supported in the graphical panes (for example, a Union query).
- The SQL statement is valid but contains syntax specific to the data connection you are using.



You can check whether a statement is valid using the **Verify SQL Syntax** button.

The dialog box displays a message with the reason that the SQL statement cannot be represented, and then asks how you want to proceed.

Note The **Query Definitions Differ** dialog box does not appear if you have hidden the Diagram and Grid panes, because the Query Designer assumes that you are editing only in the SQL pane.

Yes

Choose this button to specify that you want to accept the SQL statement, either to edit it further or to execute it. If you accept the statement, the Diagram and Grid panes appear dimmed to indicate that they do not represent the statement in the SQL pane.

No

Choose this button to discard your changes to the SQL pane.

Note If the statement is correct but not supported graphically by the Query Designer, you can execute it even though it cannot be represented in the Diagram and Grid panes. For example, if you enter a Union query, the statement can be executed but not represented in the other panes.

Save Dialog Box

Appears when you save a database diagram, one or more tables on a database diagram, or when you save your work in the table designer and that work affects other tables. This dialog box lets you confirm the tables that will be saved.

Note Although you cannot undo changes made to a table or database diagram, the changes are not saved to the database until you save the table or database diagram. You can discard any unsaved changes by closing all open database diagrams without saving them.

Yes

Saves all the tables shown in the list.

No

Cancels the save action.

Save Text File

Displays the **Save As** dialog box, prompting you for a location in which to save a text file listing the tables. This file provides a record of the tables that were affected by the changes you made.

See Also

[Save Selected Tables on a Database Diagram](#)

[Show All](#)

SQL Syntax Errors Encountered Dialog Box

Notifies you that the designer cannot parse the SQL statement in the SQL pane. This dialog box appears when you enter or edit an SQL statement in the SQL pane; and then you either switch to another pane, verify the query, or attempt to execute the query; and one of the following conditions applies:

- The SQL statement is incomplete or contains one or more syntax errors.
- The SQL statement is valid but is not supported in the graphical panes (for example, a Union query).
- The SQL statement is valid but contains syntax specific to the data connection you are using.



You can check whether a statement is valid using the **Verify SQL Syntax** button.

The dialog box displays a message with the reason that the SQL statement cannot be parsed. Choose **OK** to proceed.

Update Table Dialog Box

Allows you to specify the table to be updated. This dialog box appears if more than one table is displayed in the Diagram pane when you change a query's type to be an Update query. You can create an Update query only as part of a stored procedure.

Select the table to update, and then choose **OK**.

See Also

[Create an Update Query](#)

Validation Warnings Dialog Box

Appears if you attempt to save modifications with potentially damaging side effects— or if the database commit operation is likely to fail. This dialog box indicates what those side effects might be or why the commit operation might fail. It gives you the chance to proceed with the modification or cancel the operation.

Note This dialog box appears when you attempt to transmit your modifications to the database.

The dialog box can appear for any of these reasons:

- You might not have database permissions to commit all the modifications.
- Your modifications would result in improperly formed derived columns, default constraints, or check constraints.
- A modification to a column's data type might cause data loss.
- A modification would result in an index greater than 900 bytes.
- A modification would change a table or column contributing to a schema-bound view or user-defined function.
- A modification would result in the re-creation of a table that has one or more encrypted triggers; the triggers will be dropped.
- Your modifications will yield noteworthy settings of ANSI_NULLS or ANSI_PADDING or both for the columns within one table.

Yes

Proceeds with the operation. That is, selecting **Yes** proceeds to generate the change script or transmit the modifications to the database. Remember, even if you are transmitting the modifications to the database and you select **Yes**, the commit operation could still fail if you do not have privileges to modify the database; if your modifications would result in an index greater than 900 bytes; or if your modifications would result in an improperly formed computed column, default constraint, or check constraint.

No

Cancels the save action.

Save Text File

Displays the **Save As** dialog box, letting you specify a location for a text file containing a list of the warnings.

Check Constraints Property Page

This property page contains a set of properties for check constraints attached to the tables in a Microsoft SQL Server database. Properties applying to unique constraints appear on the Indexes/Keys property page.

Table Name

Shows the name of the selected table in your diagram. If more than one table is selected in your diagram, only the name of the first table is visible.

Selected Constraint

Shows the name of the constraint whose properties you are viewing. To view the properties of a different constraint, select a constraint from the drop-down list.

New

Choose this button to create a new constraint for the selected database table. Enter properties for the constraint. For more information, see [Attach a New Check Constraint to a Table or Column](#).

Delete

Choose this button to remove the selected constraint from the database. For more information, see [Delete a Check Constraint](#).

Constraint name

Shows the name of the constraint whose properties you are viewing. Use this control to modify the name of the constraint. For more information, see [Rename a Check Constraint](#).

Constraint expression

Shows the SQL syntax of the selected check constraint. For new constraints, you must enter the SQL syntax before exiting this box. You can also edit existing check constraints. For more information, see [Define a Check Constraint](#)

[Expression.](#)

Validation Text

(SQL Server 2000 only.) Shows the text that is displayed to the user whenever a row that violates the constraint is entered.

Check existing data on creation

When selected, this option ensures that all data that exists in the table before the constraint was created is verified against the constraint.

Enforce constraint for replication

Enforces the constraint when the table is replicated into a different database.

Enforce constraint for INSERTs and UPDATEs

Enforces the constraint when data is inserted into or updated in the table.

Columns Property Page

This property page contains a set of properties for a column within a table or for the set of output columns within a view or in-line function.

Table name

This property does not appear if you are working with a view or in-line function. Shows the name of the table containing the column whose properties you are viewing. The Table Name option is editable only within the Database Designer — not from within the Table Designer. If more than one table is selected in your diagram, only the name of the first table is visible.

Object name

This property does not appear if you are working with a table. Shows the name of the view or in-line function containing the output column whose properties you are viewing. The Object Name option is not editable.

Column Name

Shows the name of the selected column. To show properties for a different column, expand the Column Name list.

If you are working on a table, this shows the name of one of the table's columns. If you are working on a view or in-line function, this shows the name of an output column.

Expression

Applies only if you are working on a view or in-line function. Shows the database expression for the output column—the column that appears in the output of the view or in-line function. If the column corresponds directly to a source column, the expression will show the owner name, table name, and column name. For example, if the view or function includes the qty column from the sales table, the expression is `dbo.sales.qty`.

If the column corresponds to a derived column defined in the view or in-line

function, the expression shows the derivation. For example, a column called `profit` could have an expression of `revenue-expenses`.

Description

(Microsoft SQL Server 2000 only.) Shows the text description of the selected column. The value is stored as a SQL Server extended property.

Data type

Shows the data type of the selected column.

If you are working with a view or in-line function, **Data type** applies only to output columns that correspond directly to columns in the underlying table, view, or function. In other words, if the output column is derived, the **Data type** property is blank. You cannot edit this property for a view or in-line function.

Length

Applies only if you are working on a table. Shows the length of the selected column.

Nullable

Determines whether or not you can enter Null values in the column.

- **NULL** Null values are allowed.
- **NOT NULL** Null values are not allowed.

Default Value

Applies only if you are working on a table. Shows the default for this column whenever a row with no value for this column is inserted into the table. The value of this field can be either the value of a SQL Server default constraint or the name of a global constraint to which the column is bound. The drop-down list contains all global defaults defined in the database. To bind the column to a global default, select from the drop-down list. Alternatively, to create a default constraint for the column, enter the default value directly as text.

Precision

Applies only if you are working on a table. Shows the maximum number of digits for values of this column.

Scale

Applies only if you are working on a table. Shows the maximum number of digits that can appear to the right of the decimal point for values of this column.

Identity

Applies only if you are working on a table. Shows whether the column is used by SQL Server as an identity column. Possible values are:

- **No** The column is not used as an identity column.
- **Yes** The column is used as an identity column.
- **Yes (Not For Replication)** The column is used as an identity column, except while a replication agent is inserting data into the table.

Identity Seed

Applies only if you are working on a table. Shows the seed value of an identity column. This option applies only to columns whose **Identity** option is set to **Yes** or **Yes (Not For Replication)**.

Identity Increment

Applies only if you are working on a table. Shows the increment value of an identity column. This option applies only to columns whose **Identity** option is set to **Yes** or **Yes (Not For Replication)**.

Is RowGuid

Applies only if you are working on a table. Shows whether the column is used by SQL Server as a ROWGUID column. You can set this value to **Yes** only for a column that is an identity column.

Formula

(SQL Server 7.0 or higher only.) Applies only if you are working on a table. Shows the formula for a computed column.

Collation

(SQL Server 2000 only.) Applies only if you are working on a table. Shows the collating sequence that SQL Server applies by default to the column whenever the column values are used to sort rows of a query result. To use the default collating sequence for the database, choose **<database default>**.

Format

(SQL Server 2000 only.) Shows the display format for the column. To change format, select a format from the list.

Decimal Places

Shows the number of decimal places to be used for displaying values of this column. If you choose **Auto**, the number of decimal places is determined by the value you choose in **Format**.

Input Mask

(SQL Server 2000 only.) To change the input mask, click in this text box, then click the button that appears alongside it.

Caption

(SQL Server 2000 only.) Shows the text label that appears by default in forms using this column.

Indexed

(SQL Server 2000 only.) Applies only if you are working on a table. Shows whether an index exists on the column. Possible values are:

- **No** No index exists on the column.
- **Yes (duplicates OK)** A non-unique single-column index exists on the column.
- **Yes (no duplicates)** A unique single-column index exists on the column.

Hyperlink

(SQL Server 2000 only.) Indicates whether the values in this column can be interpreted as hyperlinks.

IME Mode

(SQL Server 2000 only.) Determines the IME (Input Method Editor) status of the column when users enter values into that column. Your choices are:

- **No Control** No Control is the default.
- **On** Indicates the IME is on and characters specific to Chinese, Japanese, or Korean can be entered as values for this column.
- **Off** Indicates the IME is off. When users enter values into the column, the keyboard behaves the same as English entry mode.
- **Disable** Similar to “Off”, except that “Disable” prevents users from turning the IME on.
- **Hiragana** Valid for Japanese IME only.
- **Katakana** Valid for Japanese IME only.
- **Katakana Half** Valid for Japanese IME only.
- **Alpha Full** Valid for Japanese and Korean IME only.
- **Alpha** Valid for Japanese and Korean IME only.
- **Hangul Full** Valid for Korean IME only.
- **Hangul** Valid for Korean IME only.

IME Sentence Mode

(SQL Server 2000 only.) Determines what additional IME conversion applies by default when users enter values into the column. Your choices are:

- **Phase Predict** Indicates a basic level of conversion. This is the default.
- **Plural Clause** Supports conversion using additional dictionaries with name, geographic, and postal data.
- **Conversation** Supports conversion that recognizes conversational language.
- **None** Disables character conversion.

Furigana

(SQL Server 2000 only.) Indicates a column into which the Furigana equivalent

of the text entered by the user is stored. When the user enters a value into this column, that value is stored, and in addition, the Furigana equivalent of the entered text is stored in the column named in this control.

Postal Address

(SQL Server 2000 only.) Specifies a control or field that displays either an address corresponding to an entered postal code or customer barcode data corresponding to an entered address.

Data Property Page

(Microsoft SQL Server 2000 only.) Applies to tables, views, single-statement stored procedures, and in-line functions. Shows properties about filtering, ordering, and subdatasheets.

Table Name

Appears only if you are working on a table. Shows the name of the table. If you are working in a database diagram and more than one table is selected in it, only the name of the first table is visible.

Object Name

Applies only if you are working on a view, single-statement stored procedure, or in-line function. Shows the name of the object you are working on.

Filter

Shows client-side selection criteria— criteria applied after the result set is returned from the database. This property is a string expression consisting of a WHERE clause without the WHERE keyword.

Order By

Shows client-side sorting criteria— sorting criteria applied after the result set is returned from the database. This property is a string expression that is the name of the field or fields on which you want to sort records. When you use more than one field name, separate the names with a comma (.). If you want to sort records in descending order, type DESC at the end of the string expression.

Subdatasheet Name

Specifies or determines the table or query that is bound to the subdatasheet. Choose **[Auto]** to indicate that Access will use existing relationships in the database to determine which table to bind to the subdatasheet.

Link Child Fields

Shows the list of linking fields in the subdatasheet. The fields you list here should coincide with the fields you supply in the **Link Master Fields** control.

Link Master Fields

Shows the list of linking fields or controls in the table, view, stored procedure, or in-line function containing the subdatasheet. The fields you list here should coincide with the fields you supply in the **Link Child Fields** control. Each coinciding pair of (master, child) fields do not have to have the same name, but they must contain the same kind of data and have the same or compatible data types and field sizes.

Subdatasheet Height

Shows the default height to display in the subdatasheet. If the subdatasheet contains more rows than the default height accommodates, the subdatasheet appears with a vertical scrollbar.

Subdatasheet Expanded

Indicates the saved state of all subdatasheets within the table, view, single-statement stored procedure, or in-line function. Yes means expanded, No means unexpanded.

Default View

Shows the view used to display the subdatasheet when the users opens the object or embeds it as a subform or subreport. Your choices are Datasheet, PivotTable, or PivotChart.

Function Parameters Property Page

(Microsoft SQL Server 2000 only.) Shows the parameters defined for the in-line function you are creating. Each row of the grid contains information about one parameter. Each parameter has a name, a data type, and optionally a default value.

You cannot create a new parameter with this property page; you can only modify existing parameters. To create a new parameter, enter the parameter in a Column or Criteria cell of the Grid pane or directly in the SQL pane.

Name

You cannot edit the parameter name on this property page. To change a parameter name, you must modify it wherever it appears in the Grid pane or the SQL pane. Note that the first character of every parameter name must be the at sign (@).

Data Type

The data type appears in condensed form. The condensed form includes the maximum column width in parentheses after the name of the data type. For example:

```
varchar(50)
```

Default

The default value is optional.

If you change a parameter's data type, the existing default value will be cleared.

General Property Page

Specifies options for any table, view or function appearing in the Diagram pane of a query.

Object Name

If the selected object is a table, view, or function, this control contains the object's name.

If the selected object is a subquery, this control contains the text of the subquery.

Alias

Contains the SQL alias (if any) of the selected object. You can create or modify an alias for the object by typing in this box.

Full Name

If the selected object is a table, view, or function, this control contains the object's full name (For example, Northwind.dbo.Invoices).

If the selected object is a subquery, this control contains the text of the subquery.

Indexes/Keys Property Page

This property page contains a set of properties for the indexes, primary keys, and unique constraints attached to the tables in your database diagram or in Table Designer. Indexes and constraints are not graphically represented in database diagrams.

Table name

Shows the name of the table in Table Designer or the selected table in your database diagram. If more than one table is selected in your database diagram, only the name of the first table is visible.

Selected index

Shows the name of the first index for the table in Table Designer or the selected table in your database diagram. If more than one table is selected in your database diagram, only the name of the first index for the first table is visible. To show properties for a different index, expand the drop-down list.

Type

Shows the index or key object type for the selected table: index, primary key, or unique constraint.

New

Choose this button to create a new index, key, or unique constraint for the selected database table. For more information, see [Create an Index](#).

Delete

Choose this button to remove the selected index, key, or constraint from the table. For more information, see [Delete an Index](#).

Note If you try to delete a primary key that participates in relationships, a message appears asking you if you want to delete all the relationships, too. You cannot delete a primary key without first deleting the relationships it participates

in.

Index name

Shows the name of the selected index. You can rename the index by entering a new name in this box. For more information, see [Rename an Index](#).

Column name/Order

(Microsoft SQL Server 2000 only.) Shows the columns contributing to the index, primary key, or unique constraint, along with whether each column's values are arranged in ascending or descending order within the item. You can add, change, or remove column names in this list. You can also change the ascending/descending setting for each column.

Index Filegroup

Select the name of the file group in which you want to store the selected index. You must have at least one user-defined file group for this setting to be enabled. This setting is only available for SQL Server 7.0 or higher databases. If you create a database object and do not specify its file group, SQL Server will assign it to the default file group. Initially, the default file group is the Primary file group.

For more information on creating and using file groups, see your SQL Server documentation.

Create UNIQUE

Select this option to create a unique constraint or index for the selected database table. Specify whether you are creating a constraint or index by selecting either the **Constraint** or **Index** button.

- **Ignore duplicate key** If you create a unique index, you can set this option to control how SQL Server reacts when a row whose key value equals an existing key value is inserted during a bulk insert operation. If you choose Ignore duplicate key, SQL Server will issue a warning message, ignore the offending incoming row and try to insert the remaining rows of the bulk insert operation. If you do not choose Ignore duplicate key, SQL Server will issue an error message and roll back the entire bulk insert operation.

Fill factor

Shows the fill factor that specifies how full each index page can be. If a fill factor is not specified, the database's default fill factor is used. For more information, see [Specify a Fill Factor for an Index](#).

Pad Index

If you specified a Fill Factor of more than zero percent, and you selected the option to create a unique index, you can tell SQL Server to use the same percentage you specified in Fill Factor as the space to leave open on each interior node. By default, SQL Server sets a two-row index size.

Create as CLUSTERED

Select this option to create a clustered index for the selected database table. For more information, see [Create a Clustered Index](#).

Don't automatically recompute statistics

Select this option to tell SQL Server to use previously created statistics. This choice, available only for SQL Server 7.0 and higher databases, might degrade query performance, but will accelerate the index-building operation.

For more information on this option, see your SQL Server documentation.

Validation Text

(SQL Server 2000 only.) Shows the text that is displayed to the users whenever a row that violates the index, key, or constraint is entered.

For SQL Server 6.5, the following options are exposed.

Sort Data Options

Identify how the data is ordered in the index as records are added to the index.

- **Sort data** Default. Organizes data in ascending order.
- **Data already sorted** Accepts the order of existing data.
- **Reorganize sorted data** Reorganizes the data in ascending order. Select

this option, for example, when the table becomes fragmented or to rebuild nonclustered indexes.

Duplicate Row Options

Identify how the index should handle duplicate rows.

- **Disallow duplicate rows** Default. Prevents the index from being saved if duplicate rows exist. If duplicate rows exist, an error message appears.
- **Ignore duplicate rows** Deletes duplicate rows from the index as it is being created.
- **Allow duplicate rows** Creates the index, even though duplicate rows exist.

In-Line Function Property Page

(Microsoft SQL Server 2000 only.) Contains options for controlling the behavior of the in-line function you are building or modifying.

Function

Displays the name for the current function. If you have not yet saved the function, the default name is displayed. You cannot change the name in this box. To assign a name, choose **Save** or **Save As** from the **File** menu.

Output all columns

Specifies that all columns from all tables currently displayed will be in the function's output. Choosing this option is equivalent to specifying an asterisk (*) in place of individual column names in the SQL pane.

DISTINCT values

Specifies that the query will filter out duplicate rows from the function's output. This option is useful when you are using only some of the columns from a table and those columns might contain duplicate values, or when the process of joining two or more tables produces duplicate rows in the result set. Choosing this option is equivalent to inserting the word **DISTINCT** into the **SELECT** statement in the SQL pane.

Bind to Schema

Prevents all users (including you) from deleting or modifying the structure of database objects used by this function in any way that would invalidate the function's definition.

GROUP BY extension

Specifies that additional options for in-line functions based on aggregate queries are available.

- **WITH CUBE**

Specifies that the aggregate query should produce summary values for groups specified in the GROUP BY clause. The groups are created by cross-referencing columns included in the GROUP BY clause, and then applying the query's aggregate function to produce summary values for the additional super-aggregate rows. The WITH CUBE option is multidimensional, creating summaries from all combinations of aggregate functions and columns in the query.

- **WITH ROLLUP**

Specifies that the aggregate query should produce summary values for the groups specified in the GROUP BY clause. The WITH ROLLUP option is similar to the WITH CUBE option, but creates only one type of summary for each combination of column and aggregate function.

- **ALL**

Specifies that the aggregate query will include any duplicate rows in the summary.

Owner

Shows the name of the function's owner. The owner name is either a SQL Server role or SQL Server user. The drop-down list contains all the users and roles defined in the database. Within the drop-down list, the users and roles have different icons; the role icon shows two faces, the user icon shows only one.

TOP

(For SQL Server 7.0 or higher databases.) Specifies that the in-line function will include a TOP clause, which returns only the first n rows or first n percentage of rows in the result set. The default is that the function returns the first 10 rows in the result set.

Use this box to specify a different number of rows to return or to specify a percentage.

- **PERCENT**

(For SQL Server 7.0 or higher databases.) Specifies that the function will include a TOP clause, returning only the first n percentage of rows in the result set.

- **WITH TIES**

(For SQL Server 7.0 or higher databases.) Specifies that the function will include a WITH TIES clause. WITH TIES is useful if a function includes an ORDER BY clause and a TOP clause based on percentage. If this option is set, and if the percentage cutoff falls in the middle of a set of rows with identical values in the ORDER BY clause, the view is extended to include all such rows.

SQL Comment

To enter text in this field, you can type directly into the field, or you can enter a comment in the SQL pane. When you enter text directly into this field, you simply type the comment. But when you enter comment text in the SQL pane, you must use the comment delimiter characters. For example:

```
/* This is comment text entered in the SQL pane */  
This is comment text entered directly in the SQL Comment field.
```

Description

(SQL Server 2000 only.)

This field is distinguished from the **SQL Comment** field in two ways. First, values of **Description** are stored in SQL Server as extended properties. Second, you cannot enter values of the **Description** from the SQL pane.

Join Line Property Page

Specifies options for joining tables.

By default, related tables are joined using an inner join that creates a result set based on rows containing matching information in the join columns. By setting options in the Join Line Property Page, you can specify a join based on a different operator, and you can specify an outer join.

For more information about join types, see [Types of Joins](#).

Table

The names of the tables, views or functions involved in the join. You cannot change the names of the tables here.

Column

The names of the columns used for joining the tables. The operator in the Operator list specifies the relationship between the data in the columns. You cannot change the names of the columns here— this information is displayed for information only.

Join operator

Specifies the operator used to relate the join columns. To specify an operator other than equal (=), select it from the list. When you close the property page, the operator you selected will appear in the diamond graphic of the join line, as in the following:



Include rows

Specifies whether unmatched rows appear in the datasheet.

All rows from <table1>

Specifies that all the rows from the left table appear in the output, even if there are no corresponding matches in the right table. Columns with no matching data in the right table appear as null. Choosing this option is equivalent to specifying LEFT OUTER JOIN in the SQL statement.

All rows from <table2>

Specifies that all the rows from the right table appear in the output, even if there are no corresponding matches in the left table. Columns with no matching data in the left table appear as null. Choosing this option is equivalent to specifying RIGHT OUTER JOIN in the SQL statement.

Selecting both All rows from <table1> and All rows from <table2> is equivalent to specifying FULL OUTER JOIN in the SQL statement.

When you select an option to create an outer join, the diamond graphic in the join line changes to indicate that the join is a left outer, right outer, or full outer join.

Note The words "left" and "right" do not necessarily correspond to the position of tables in the Diagram pane. "Left" refers to the table whose name appears to the left of the keyword JOIN in the SQL statement, and "right" refers to the table whose name appears to the right of the JOIN keyword. If you move tables in the Diagram pane, you do not change which table is considered left or right.

Lookup Property Page

(Microsoft SQL Server 2000 only.) This property page contains a set of properties pertaining to lookup fields. This property page applies to tables, views, and in-line functions.

Table Name

Appears only if you are working on a table. Shows the name of the table. If you are working in a database diagram and more than one table is selected in it, only the name of the first table is visible.

Object Name

Appears only if you are working on a view or in-line function. Shows the name of the view or function. You cannot edit this property.

Column Name

If you are working on a table, shows the name of a column within that table. If you are working on a view or in-line function, shows the name of an output column for that view or function. To inspect the lookup properties for a different column, choose from the drop-down list.

Display Control

Shows the default control for displaying values of this column when you include it on a form.

Row Source Type

Works together with the **Row Source** property. Use this property to specify the type of row source (table/query, value list, field list, or Visual Basic function) and then specify the actual row source in the **Row Source** property.

Row Source

If **Row Source Type** is set to Table/Query, shows the name of the table, query,

or SQL statement. Although stored procedures are not displayed, you can enter the name of a single-statement stored procedure that does not modify data.

If **Row Source Type** is set to Value List, shows the entries in the list, separated by semicolons.

If **Row Source Type** is set to Field List, shows the name of the table or query.

If **Row Source Type** is set to Function Name, **Row Source** is blank.

Bound Column

Shows which column is bound to the underlying field for a bound multicolumn list box or combo box.

Column Count

Shows the number of columns on a list box or combo box.

Column Widths

Shows the width of each column, separated by semicolons. To hide the bound column from the display, use a width value of zero.

List Rows

Shows the maximum number of rows to display in the list box portion of a combo box.

List Width

Shows the width of the list box portion of a combo box.

Limit To List

Determines whether a combo box accepts any entered text or only text that matches one of the values in the list.

Query Property Page

Contains options for controlling the behavior of the query you are building or modifying.

Query name

Displays the name for the current query. You cannot change the query name in this box.

Output all columns

Specifies that all columns from all tables in the current query will be in the result set. Choosing this option is equivalent to specifying an asterisk (*) in place of individual column names after the SELECT keyword in the SQL statement.

DISTINCT values

Specifies that the query will filter out duplicates in the result set. This option is useful when you are using only some of the columns from the table or tables and those columns might contain duplicate values, or when the process of joining two or more tables produces duplicate rows in the result set. Choosing this option is equivalent to inserting the word DISTINCT into the statement in the SQL pane.

Destination table

Specifies the name of the table into which you are inserting rows. This list appears if you are creating an Append query or Make-Table query. For an Append query, select a table name from the list.

For a Make-Table query, type the name of the table to create. The Query Designer does not check whether the name is already in use or whether you have permission to create the table. To create a destination table in another database, specify a fully qualified table name, including the name of the target database, the owner (if required), and the name of the table.

GROUP BY extension

Specifies that additional options for queries based on aggregate queries are available.

- **WITH CUBE**

Specifies that the aggregate query should produce summary values for groups specified in the GROUP BY clause. The groups are created by cross-referencing columns included in the GROUP BY clause, and then applying the query's aggregate function to produce summary values for the additional super-aggregate rows. The WITH CUBE option is multidimensional, creating summaries from all combinations of aggregate functions and columns in the query.

- **WITH ROLLUP**

Specifies that the aggregate query should produce summary values for the groups specified in the GROUP BY clause. The WITH ROLLUP option is similar to the WITH CUBE option, but creates only one type of summary for each combination of column and aggregate function.

- **ALL**

Specifies that the aggregate query will include any duplicate rows in the summary.

TOP

(For SQL Server 7.0 or higher databases.) Specifies that the query will include a TOP clause, which returns only the first n rows or first n percentage of rows in the result set. The default is that the query returns the first 10 rows in the result set.

Use this box to specify a different number of rows to return or to specify a percentage.

- **PERCENT**

(For SQL Server 7.0 or higher databases.) Specifies that the query will include a TOP clause, returning only the first n percentage of rows in the result set.

- **WITH TIES**

(For SQL Server 7.0 or higher databases.) Specifies that the query will include a WITH TIES clause. WITH TIES is useful if a query includes an ORDER BY clause and a TOP clause based on percentage. If this option is set, and if the percentage cutoff falls in the middle of a set of rows with identical values in the ORDER BY clause, the query extends the percentage until all such rows are included.

SQL Comment

To enter text in this field, you can type directly into it, or you can enter a comment in the SQL pane. When you enter text directly into this field, you simply type the text. But when you enter comment text in the SQL pane, you must use the comment delimiter characters. For example:

```
/* This is comment text entered in the SQL pane */  
This is comment text entered directly in the SQL Comment field.
```

Relationships Property Page

This property page contains a set of properties for the relationships between the tables in your database.

Table name

Shows the name of the table in Table Designer or the selected table in your database diagram. If more than one table is selected in your database diagram, only the name of the first table is visible.

Selected relationship

Shows the name of the selected relationship in your database diagram. If more than one relationship is selected in your database diagram, only the name of the first relationship in your selection is visible. Expand the list to view or modify the properties of a different relationship.

Each entry in the drop-down list is preceded by an icon. A key icon indicates that the table participates in the relationship as the referred-to table. An infinity icon indicates that the table participates as the referring table. (The referring table has the relationship's foreign-key constraint.)

New

Choose this button to create a new relationship for the selected database table. For more information, see [Create a Relationship Between Tables](#).

Delete

Choose this button to remove the selected relationship from the database. For more information, see [Delete a Relationship](#).

Relationship name

Shows the name of the selected relationship. You can rename the relationship by entering a new name in this box. For more information, see [Rename a Relationship](#).

Primary key table

Shows the name of the primary key table in the relationship, followed by the columns that make up the primary key. For information about changing the primary key, see [Rename a Primary Key](#).

Foreign key table

Shows the name of the foreign key table in the relationship, followed by the columns that make up the foreign key. For information about changing the foreign key, see [Modify a Foreign Key](#).

Check existing data on creation

Applies the constraint to data that already exists in the database when the relationship is added to the foreign key table.

Enforce relationship for replication

Applies the constraint when the foreign key table is copied to a different database.

Enforce relationship for INSERTs and UPDATEs

Applies the constraint to data that is inserted into, deleted, or updated in the foreign key table. Also prevents a row in the primary key table from being deleted when a matching row exists in the foreign key table.

- **Cascade Update Related Fields**

(Microsoft SQL Server 2000 only.) Instructs the database to automatically update foreign-key values of this relationship whenever the primary-key value is updated.

- **Cascade Delete Related Fields**

(SQL Server 2000 only.) Instructs the database to automatically delete rows of the foreign-key table whenever the referred-to row of the primary-key table is deleted.

Validation Text

(SQL Server 2000 only.) Shows the text that is displayed to the users whenever a row that violates the foreign-key constraint is entered.

Stored Procedure Parameters Property Page

Shows the parameters defined for the stored procedure you are creating. Each row of the grid contains information about one parameter. Each parameter has a name, a data type, and optionally a default value.

You cannot create a new parameter with this property page; you can only modify existing parameters. To create a new parameter, enter the parameter in a Column or Criteria cell of the Grid pane or directly in the SQL pane.

Name

You cannot edit the parameter name on this property page. To change a parameter name, you must modify it wherever it appears in the Grid pane or the SQL pane. Note that the first character of every parameter name must be the at sign (@).

Data Type

The data type appears in condensed form. The condensed form includes the maximum column width in parentheses after the name of the data type. For example:

```
varchar(50)
```

Default

The default value is optional.

If you change a parameter's data type, the existing default value will be cleared.

Stored Procedure Property Page

(Microsoft SQL Server 2000 only.) Contains options for controlling the behavior of the single-statement stored procedure you are building or modifying.

Procedure

Displays the name for the current stored procedure. If you have not yet saved the procedure, the default name is displayed. You cannot change the name in this box. To assign a name, choose **Save** or **Save As** from the **File** menu.

Output all columns

Specifies that all columns from all tables currently displayed will be in the procedure's output. Choosing this option is equivalent to specifying an asterisk (*) in place of individual column names in the SQL pane.

DISTINCT values

Specifies that the query will filter out duplicate rows from the procedure's output. This option is useful when you are using only some of the columns from a table and those columns might contain duplicate values, or when the process of joining two or more tables produces duplicate rows in the result set. Choosing this option is equivalent to inserting the word **DISTINCT** into the statement in the SQL pane.

Set NOCOUNT on

Prevents SQL Server from including in the query result a message indicating the number of rows affected by the Transact-SQL statement in the body of the stored procedure. To see the message, clear this check box.

GROUP BY extension

Specifies that additional options for stored procedures based on aggregate queries are available.

- **WITH CUBE**

Specifies that the aggregate query should produce summary values for groups specified in the GROUP BY clause. The groups are created by cross-referencing columns included in the GROUP BY clause, and then applying the query's aggregate function to produce summary values for the additional super-aggregate rows. The WITH CUBE option is multidimensional, creating summaries from all combinations of aggregate functions and columns in the query.

- **WITH ROLLUP**

Specifies that the aggregate query should produce summary values for the groups specified in the GROUP BY clause. The WITH ROLLUP option is similar to the WITH CUBE option, but creates only one type of summary for each combination of column and aggregate function.

- **ALL**

Specifies that the aggregate query will include any duplicate rows in the summary.

Owner

Shows the name of the procedure's owner. The owner name is either a SQL Server role or SQL Server user. The drop-down list contains all the users and roles defined in the database. Within the drop-down list, the users and roles have different icons; the role icon shows two faces, the user icon shows only one.

TOP

(For SQL Server 7.0 or higher databases.) Specifies that the stored procedure will include a TOP clause, which returns only the first n rows or first n percentage of rows in the result set. The default is that the procedure returns the first 10 rows in the result set.

Use this box to specify a different number of rows to return or to specify a percentage.

- **PERCENT**

(For SQL Server 7.0 or higher databases.) Specifies that the procedure will include a TOP clause, returning only the first n percentage of rows in the result

set.

- **WITH TIES**

(For SQL Server 7.0 or higher databases.) Specifies that the procedure will include a **WITH TIES** clause. **WITH TIES** is useful if a procedure includes an **ORDER BY** clause and a **TOP** clause based on percentage. If this option is set, and if the percentage cutoff falls in the middle of a set of rows with identical values in the **ORDER BY** clause, the view is extended to include all such rows.

SQL Comment

To enter text in this field, you can type directly into it, or you can enter a comment in the SQL pane. When you enter text directly into this field, you simply type the text. But when you enter comment text in the SQL pane, you must use the comment delimiter characters. For example:

```
/* This is comment text entered in the SQL pane */  
This is comment text entered directly in the SQL Comment field.
```

Description

(SQL Server 2000 only.)

This field is distinguished from the **SQL Comment** field in two ways. First, values of **Description** are stored in SQL Server as extended properties. Second, you cannot enter values of the **Description** from the SQL pane.

Tables Property Page

This property page contains a set of properties for the tables in your database.

Selected table

Shows the name of the table in Table Designer or the selected table in your database diagram. If more than one table is selected in your database diagram, only the name of the first table in your selection is visible. Expand the list to choose a different table whose properties you want to inspect or modify.

Owner

Shows the name of the table's owner. The owner name is either a Microsoft SQL Server role or SQL Server user. The drop-down list contains all the users and roles defined in the database. Within the drop-down list, the users and roles have different icons; the role icon shows two faces, the user icon shows only one.

Table name

Shows the name of the selected table. To rename the table, enter a new name in this box. For more information, see [Rename a Table in a Database Diagram](#).

Table Identity Column

Shows the column used by SQL Server as the table's identity column. To change the identity column, choose from the drop-down list. Within the drop-down list, you can choose the blank entry to indicate that the table has no identity column.

Table ROWGUID Column

(SQL Server 7.0 or higher only.) Shows the column used by SQL Server as the table's ROWGUID column. To change the ROWGUID column, choose from the drop-down list. Within the drop-down list, you can choose the blank entry to indicate that the table has no ROWGUID column.

Table Filegroup

Select the name of the file group in which you want to store the selected table data. You must have at least one user-defined file group for this setting to be enabled. This setting is only available for SQL Server 7.0 or higher databases. If you create a database object and do not specify its file group, SQL Server will assign it to the default file group. Initially, the default file group is the Primary file group.

For more information on creating and using file groups, see your SQL Server documentation.

Text Filegroup

Select the name of the file group in which you want to store the text and images from the selected table. You must have at least one user-defined file group for this setting to be enabled. This setting is only available for SQL Server 7.0 or higher databases. If you create a database object and do not specify its file group, SQL Server will assign it to the default file group. Initially, the default file group is Primary file group.

For more information on creating and using file groups, see your SQL Server documentation.

Description

You can enter any text in this field. The text that you enter is implemented as a SQL Server 2000 extended property. This field is available only when you are connected to a SQL Server 2000 database.

View Property Page

Contains options for controlling the behavior of the view you are building or modifying.

View name

Displays the name for the current view. If you have not yet saved your view, the default name is displayed. You cannot change the name in this box. To assign a name, choose **Save** or **Save As** from the **File** menu.

Output all columns

Specifies that all columns from all tables currently displayed will be in the view. Choosing this option is equivalent to specifying an asterisk (*) in place of individual column names.

DISTINCT values

Specifies that the view will filter out duplicates. This option is useful when you are using only some of the columns from a table and those columns might contain duplicate values, or when the process of joining two or more tables produces duplicate rows in the result set. Choosing this option is equivalent to inserting the word **DISTINCT** into the statement in the SQL pane.

Bind To Schema

Prevents users from modifying the underlying objects that contribute to this view in any way that would invalidate the view definition.

Update using view rules

Indicates that all updates and insertions to the view will be translated by Microsoft Data Access Components (MDAC) into SQL statements that refer to the view, rather than into SQL statements that refer directly to the view's base tables. (In some cases, MDAC manifests view update and view insert operations as updates and inserts against the view's underlying base tables. By selecting **Update using view rules**, you can ensure that MDAC generates update and

insert operations against the view itself.)

CHECK Option

Indicates that whenever a user modifies a row of the view, the database checks that the added or modified data satisfies the WHERE clause of the view definition.

GROUP BY extension

Specifies that additional options for views based on aggregate queries are available.

WITH CUBE

Specifies that the aggregate query should produce summary values for groups specified in the GROUP BY clause. The groups are created by cross-referencing columns included in the GROUP BY clause, and then applying the query's aggregate function to produce summary values for the additional super-aggregate rows. The WITH CUBE option is multidimensional, creating summaries from all combinations of aggregate functions and columns in the query.

- **WITH ROLLUP**

Specifies that the aggregate query should produce summary values for the groups specified in the GROUP BY clause. The WITH ROLLUP option is similar to the WITH CUBE option, but creates only one type of summary for each combination of column and aggregate function.

- **ALL**

Specifies that the aggregate query will include any duplicate rows in the summary.

Owner

Shows the name of the view's owner. The owner name is either a SQL Server role or SQL Server user. The drop-down list contains all the users and roles defined in the database. Within the drop-down list, the users and roles have different icons; the role icon shows two faces, the user icon shows only one.

TOP

(For SQL Server 7.0 or higher databases.) Specifies that the query will include a TOP clause, which returns only the first n rows or first n percentage of rows in the result set. The default is that the query returns the first 10 rows in the result set.

Use this box to specify a different number of rows to return or to specify a percentage.

- **PERCENT**

(For SQL Server 7.0 or higher databases.) Specifies that the query will include a TOP clause, returning only the first n percentage of rows in the result set.

- **WITH TIES**

(For SQL Server 7.0 or higher databases.) Specifies that the view will include a WITH TIES clause. WITH TIES is useful if a view includes an ORDER BY clause and a TOP clause based on percentage. If this option is set, and if the percentage cutoff falls in the middle of a set of rows with identical values in the ORDER BY clause, the view is extended to include all such rows.

SQL Comment

To enter text in this field, you can type directly into it, or you can enter a comment in the SQL pane. When you enter text directly into this field, you simply type the text. But when you enter comment text in the SQL pane, you must use the comment delimiter characters. For example:

```
/* This is comment text entered in the SQL pane */  
This is comment text entered directly in the SQL Comment field.
```

Description

(SQL Server 2000 only.)

This field is distinguished from the **SQL Comment** field in two ways. First, values of **Description** are stored in SQL Server as extended properties. Second, you cannot enter values of the **Description** from the SQL pane.

<0s> cannot be used in this query type.

The action you are attempting is not permitted with the type of query you have selected. Verify the syntax in your query or change the query type.

For more information, see [About Supported Query Types](#).

<0s> in expression is not part of the query.

An expression within your query is not properly formed. This often occurs when the criteria expression that you entered doesn't match the available columns for the input sources. Clear the value out and then type in a column reference that matches your input sources.

For more information, see [About Expressions in Queries](#).

<0s> support not available in this server version.

You have attempted to use a feature that isn't supported by your server software.

A blank constraint expression is not allowed.

You must enter a constraint in the **Constraints Expression** dialog box before continuing to define properties for your table.

For more information, see [Create a Unique Constraint](#).

A database diagram with that name already exists in the database.

A database diagram of that name already exists in the database. Choose Yes to replace the existing database diagram with the one you are saving.

A number of rows were affected.

The query executed successfully. The specified rows were affected by a make table process.

A pair of matching columns is required to create a relationship.

You must select matching columns from both tables participating in a foreign key relationship. The columns you have selected don't match, so they cannot be related.

Choose two columns that match, then attempt to establish the relationship. In order to match, there must be the same number of columns and each partner in a pair of columns must have a comparable data type.

For more information, [Create a Relationship Between Tables](#).

A primary key can't be created on column '<0s>' because it allows null values.

The primary key is used to relate the table to foreign keys in other tables. All columns participating in a primary key must contain defined values other than NULL.

To create a primary key on the selected column, first clear the Allow Nulls check box for the column.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

A primary key or index cannot be created on columns with a datatype of <Os>.

A column's data type defines what kind of information can be stored in a column. Some data types such as text or image cannot be used in an index because servers will not allow indexes on these types of data.

For more information, see [About Indexes](#), [Create a Unique Index](#), and [Define a Primary Key Constraint](#).

A primary key or index cannot have more than 16 columns.

Microsoft SQL Server does not allow more than 16 columns in an index. Because a primary key has an associated index, there can be no more than 16 columns in a primary key.

For more information, see [About Indexes](#) and [Define a Primary Key Constraint](#).

A primary key or unique constraint must be defined for table '<0s>' before it can participate in a relationship.

In order to create a foreign key relationship, at least one of the two tables must have a primary key or unique constraint defined. To create the relationship you are attempting to create, first create a primary key for one of the tables.

For more information, see [Define a Primary Key Constraint](#) and [Create a Relationship Between Tables](#).

A relationship cannot contain more than '<0d>' columns.

Select fewer than 16 columns when defining the foreign key relationship.

For more information, see [Create a Relationship Between Tables](#).

Adding a formula to the selected column will delete all indexes and relationships it participates in.

The column you are modifying participates in one or more relationships or contributes to one or more indexes. When you modify the column, it will become a computed column, which cannot participate in relationships or contribute to indexes. Choose Yes to proceed with the operation and delete the indexes and relationships. Choose No to cancel the operation.

Adding a formula to the selected column will delete all relationships it participates in as a foreign key.

The column to which you are adding a formula contributes to one or more foreign keys. The modification will delete the attendant foreign-key relationships. Choose Yes to proceed with the modification and delete the relationships. Choose No to cancel the modification.

ALIAS name is already being used.

Alias names must be unique. Enter a different alias name.

For more information, see [Create Table Aliases in a Query](#) and [Create Column Aliases in a Query](#).

ALL cannot be used with CUBE or ROLLUP.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Ambiguous outer join (+) operator.

The Query Designer requires that you follow specific rules when designing queries. The Query Designer is unable to process the outer join operator you have entered. Check the syntax in your SQL statement.

For more information, see [Create Outer Joins in a Query](#).

An expression cannot be used as a parameter value.

An expression appears as a parameter value. Parameter values must be scalar quantities or simple text. Supply a text value or number for the quantity, as appropriate.

An index can't exist on a blank column.

An index must contain at least one column, and that column must contain unique data values if you are creating a unique index. The column names selected must be entered under Column name in the Selected index box.

For more information, see [About Indexes](#) and [Create an Index](#).

Another user modified this database diagram while you were working on it.

Since you opened the database diagram and began working on it, another user has saved modifications to it in the database. If you save your modifications, you will overwrite that user's changes.

For more information, see [About Multiuser Environments](#).

Appropriate SQL cannot be generated.

The Query Designer requires that you follow specific rules when designing queries. This error may appear if you have used the Diagram pane or Grid pane to create a query that cannot be expressed in an SQL statement. Verify the query options you have selected.

For more information, see [About Creating Queries](#).

Are you sure you want to delete the selected relationship from your database?

Answering YES will delete the selected relationship.

For more information about relationships, see [About Table Relationships](#) and [Delete a Relationship](#).

Are you sure you want to permanently delete table '<0s>' from your database?

This will permanently delete the table and all its data from the database.

For more information, see [Delete a Table from a Database Diagram and the Database](#).

Are you sure you want to permanently delete the selected tables from your database?

The table and all its data will be deleted from the database.

For more information, see [Delete a Table from a Database Diagram and the Database](#).

Asterisk (*) may not be used in an indexed view.

Because an indexed view is stored on disk, the view definition must explicitly refer to the columns to be included in the view. Replace the asterisk character (*) with the columns you want to include in the view.

Bad top value.

The top value in your query is invalid or out of range. The value for the top clause must be an integer value if the PERCENT clause is not used. To use the PERCENT clause, just type in the word PERCENT after the numeric value in the cell.

For more information, see [Rules for Entering Search Values](#).

Both sides of a relationship must have the same number of columns.

When creating a composite foreign key relationship, both sides of the relationship must have the same number of columns. Make sure the number of columns you've selected matches the number of columns in the target table.

For more information, see [Create a Relationship Between Tables](#).

Cannot assign alias to this field.

An alias can only be created for a suitable column or table. The field you have chosen is not valid. Verify that the settings for the field are correct.

For more information, see [Create Column Aliases in a Query](#) or [Create Table Aliases in a Query](#).

Cannot convert entry to valid date/time.

The Query Designer requires that you follow specific rules when designing queries. Verify that the entry is correct.

For more information, see [About Creating Queries](#).

Cannot convert to proper type.

The Query Designer requires that you follow specific rules when converting values. The query cannot be converted. Verify that the information in the query and the query type are appropriate.

For more information, see [Functions for Expressions](#).

Cannot delete rows with unknown keys.

The row you've attempted to delete has an unknown key. This row cannot be deleted at this time.

Cannot edit rows with unknown keys.

The row you've attempted to edit has an unknown key.

Cannot edit this cell.

The cell you've selected cannot be edited. It may be the result of a calculation or part of a joined table.

Cannot filter this expression.

The expression you've entered cannot be filtered. This typically occurs when the user types in a criteria against the * or a criteria against a field that is blank in the Filter By Form or Server Filter By Form window. The user should only enter criteria against fields that are already in the Filter By Form or Server Filter By Form window.

Cannot insert into this expression.

You cannot insert data into the selected expression. This function is not supported at this time.

For more information, see [Create an Append Query](#).

Cannot open encrypted <0s> <1s>.

After a view has been encrypted and closed it can't be opened again.

Cannot put expression on select list.

There are limitations to what expressions are allowed in the select list. The expression you are trying to add may be invalid. Verify that the syntax in the expression is correct. It is possible that the type of expression is invalid.

For more information, see [About Expressions in Queries](#).

Cannot update this expression.

The selected expression cannot be updated.

For more information, see [About Expressions in Queries](#).

Cannot use column whose data type is image or text in this context.

The Query Designer requires that you follow specific rules when designing queries.

For more information, see [About Creating Queries](#).

Cannot use long data type in this context.

The Query Designer requires that you follow specific rules when designing queries.

For more information, see [About Creating Queries](#).

Changing a column data type results in an index that is too large.

The column whose data type you are changing contributes to one or more indexes. The new data type enlarges the column, making one or more of that column's indexes too large. Choose Yes to proceed with the modification and delete the index or indexes. Choose No to cancel the modification.

Changing the column to the selected data type will delete its indexes and all relationships it participates in.

The column whose data type you are changing participates in indexes or relationships or both. But the new data type you supplied cannot accommodate participation in indexes or relationships. If you proceed, the data type will change, but the existing indexes and relationships will be lost.

Changing the column to the selected data type will delete its indexes.

The column whose data type you are changing contributes to one or more indexes. The modification will delete the indexes. Choose Yes to proceed with the modification and delete the indexes. Choose No to cancel the modification.

Changing the column to the selected data type will delete relationships.

The columns whose data types you are changing participate in foreign-key relationships. If the data types are changed, the relationships will be deleted. Choose Yes to proceed with the operation and delete the relationships. Choose No to stop the operation.

Note Choosing No does not cancel the portion of the operation that has already been completed in the database.

Changing this alias will result in a duplicate output column name.

The modification you are trying to make will yield a query in which several output columns have the same name. You must explicitly supply unique aliases for each output column.

Collate clause may not be used in an indexed view.

The COLLATE clause appears within the definition of an indexed view, but an indexed view definition cannot contain the COLLATE clause. Modify the view definition by either removing the COLLATE clause or removing the indexes from the view.

Column '<0s>' is a computed column, and cannot contribute to foreign keys.

You tried to use a column with a formula in a foreign key. Either clear the column's **Formula** property or remove the column from the foreign key.

Column '<0s>' is a computed column, and cannot participate in indexes or relationships.

You tried to use a column with a formula in an index or relationship. In Microsoft SQL Server 7.0 databases, a computed column cannot participate in an index or a relationship. In SQL Server 2000, a computed column can participate in indexes, but not in relationships. Either clear the column's **Formula** property or remove the column from the index or relationship.

Column '<0s>' is used to enforce the full-text key on table <1s> and must be <2d> bytes or less.

You are performing an operation that will enlarge a column used as a key for full-text indexing of this table beyond the maximum size for such a column. Choose Yes to proceed with the operation and disable full-text indexing on the table. Choose No to cancel the operation.

Column '<0s>' participates in full-text indexing on table '<1s>'. After this change, the column will no longer participate in the table's full-text index.

You are performing an operation that disallows full-text searching of the data in this column. Choose Yes to proceed with the operation and disable full-text searches of this column.

Even if you choose Yes, other columns that participate in this table's full-text indexing will be unaffected. That is, you will still be able to perform full-text searches of the data in those columns.

Column '<1s>' in table '<0s>' participates in index '<2s>'. Columns with a datatype of '<3s>' cannot participate in indexes or primary keys.

You have attempted to change the datatype of a column that is part of an index or a primary key. Indexes and primary keys cannot be created for columns with datatypes such as text or image. To change the datatype, first remove the index or primary key.

For more information see [About Indexes](#), [Create a Unique Index](#), and [Define a Primary Key Constraint](#).

Column cannot be updated.

You cannot update values in this column— perhaps because it is a computed column, and identity column, or a column whose values are ROWGUIDS.

Column has no name and will be deleted.

Because the column-name field is blank, the column will be deleted. Choose Yes to delete the column. Choose No to return to the Table Designer or database diagram to supply a column name.

Column is the full-text filter for columns that participate in full-text indexing on a table.

You are trying to delete a column or change its data type, but that column is the full-text filter for one or more other columns for which a full-text index exists. If you proceed with the operation, the columns will no longer participate in the full-text index.

If you later want to reestablish full-text indexing for these columns, you must choose or create a new full-text filter field, and you must add these columns back to the full-text index. For more information about accomplishing these tasks, see the topics on full-text indexing in the Microsoft SQL Server documentation.

Columns in this expression have incompatible collations.

An expression you supplied in an indexed view definition uses two columns with different collation settings, so a collation clause has been added to one of the columns. But an indexed view definition cannot contain the collation clause. Modify the view definition by either removing the expression with the incompatible columns or remove the indexes from the view.

Comment exceeds n bytes.

The comment exceeds the maximum length allowed. Enter a shorter comment.

Connection failure.

The connection to the database is no longer valid. Try to reconnect before proceeding.

Data source alias is read only.

Read-only values cannot be edited. To use this data source, you need a write-enabled version of the data source available to the Query Designer.

Data source base name is read only.

Read-only values cannot be edited. To use this data source, you need a write-enabled version of the data source available to the Query Designer.

Data type error in expression.

Expressions with different data types can generate errors. To work with different data types in a single expression, you need to convert to the same data type.

For more information, see [Structure of Expressions](#).

Data type mismatch - no conversion possible.

To work with different data types in a single expression, you need to convert to the same data type. The selected data types cannot be converted.

For more information, see [About Expressions in Queries](#).

Data type mismatch - use the Transact SQL CONVERT function.

To work with different data types in a single expression, you need to convert to the same data type. One of the data types must be converted before you can continue.

For more information, see [About Expressions in Queries](#).

Database Designer Error

No specific help is available for this error. If the text in the error message does not provide enough information for you to solve the problem, please consider the following options:

- Verify that your settings are correct and try the action again.
- Use the index to search for keywords related to the error message.
- If the problem is due to external applications or software, consult the the Microsoft SQL Server documentation or other documentation for that area. For example, errors may be caused by your database software, your server setup, or even the network you are using.
- For database modification errors, such as invalid parameters in a database table, consult your database software for details.
- Check the Readme for late-breaking information concerning the error.

Database Server Version Runtime Error.

The database server has returned a run-time error. Refer to the Microsoft SQL Server documentation or error documentation for your database.

Default Value property cannot be set on column '<0s>' because it is a computed column.

You tried to set a default value on a column that has a formula. Remove the column's formula before setting a default.

DELETE statement conflicted with COLUMN REFERENCE constraint.

A conflict occurred while your query attempted to execute. The Query Designer has tried to locate the direct source of the conflict. Please check the syntax of your query to attempt to resolve the conflict.

Deleting the selected columns will also delete relationships.

The columns you have chosen to delete have existing relationships. Deleting these columns will delete the relationships associated with them.

For more information about relationships, see [About Table Relationships](#) and [Delete a Relationship](#).

Do you want to save changes in this database diagram?

You have made changes in this database diagram. You will lose the changes if you close this database diagram without saving. Do you want to save changes?

For more information about saving database diagrams, see [Save Selected Tables on a Database Diagram](#) and [Save an Entire Database Diagram](#).

Do you want to save changes in this table?

You have made changes in this table. You will lose the changes if you close this table without saving. Do you want to save changes?

For more information about working with tables, see [About Tables](#).

Do you want to suppress further error messages telling you why records can't be pasted?

If you answer Yes, you will not receive messages explaining why records cannot be pasted correctly.

Editing this default will permanently unbind the named default.

Changing from a named default to a default constraint is not allowed on Microsoft SQL Server. Refer to the Microsoft SQL Server documentation for more information.

Edits not allowed. HAVING clause not allowed in an indexed view.

You cannot use the HAVING clause in the view definition of an indexed view.
You can remove the indexes or you can remove the HAVING clause.

Enter an expression in the Column cell first.

Enter an expression in the column cell.

For more information, see [About Expressions in Queries](#).

Error after function '<0s>'

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error after predicate near '<0s>'

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error before EXISTS clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error before EXISTS clause: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error before FROM clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error following UNION operator.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in column list.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in column list: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in destination table specification.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For best results, do not include periods in the destination table specification.

For more information, see [About Creating Queries](#).

Error in FROM clause near '<0s>'

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in GROUP BY clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Aggregate Functions](#).

Error in GROUP BY clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Aggregate Functions](#).

Error in HAVING clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in HAVING clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Summarize or Aggregating Values for All Rows in a Query](#).

Error in INSERT statement.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Error in INSERT statement: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Error in join expression.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in join expression: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in list of function arguments: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in list of values in IN clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in list of values in IN clause: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in list of values.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in list of values: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in ON clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in optional FROM clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in ORDER BY clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in ORDER BY clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in SELECT clause: alias '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in SELECT clause: alias not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in SELECT clause: expression near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in set list in UPDATE clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in table name or view name in DELETE clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query. Be sure to use Quoted Identifiers as needed.

For more information, see [About Creating Queries](#).

Error in table name or view name in DELETE clause: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query. Be sure to use Quoted Identifiers as needed.

For more information, see [About Creating Queries](#)

Error in table name or view name in INSERT clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query. Be sure to use Quoted Identifiers as needed.

Error in table name or view name in INSERT clause: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query. Be sure to use Quoted Identifiers as needed.

Error in table name or view name in UPDATE clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query. Be sure to use Quoted Identifiers as needed.

For more information, see [About Creating Queries](#).

Error in text following query statement: '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error in values list in INSERT INTO clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Error in values list in INSERT INTO clause: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Make sure the number of items in the Column list matches the number of items in the Values list.

For more information, see [About Creating Queries](#).

Error in WHERE clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Error loading table.

The table could not be loaded into local memory for the Database Designer or Table Designer. Some possible causes of this error are:

- An error occurred reading the table's metadata from the database
- The table's metadata is corrupted

For more information about how the Database Designer and Table Designer use local memory, see [Interactions Among Database Diagrams and Table Design Windows](#).

Error validating constraint.

The text of the **Constraint Expression** property does not conform to the constraint syntax recognized by Microsoft SQL Server. Some common reasons why a constraint expression can violate the required syntax are:

- The expression does not resolve to a value of TRUE or FALSE.
- The expression misspells a column name, or refers to a column in another table.
- The expression misspells the name of a function or database constant.

For a complete description of the syntax for constraint expressions in SQL Server 2000, see the SQL Server documentation.

Error validating the default for a column.

The text of the **Default Value** property is inconsistent with one or more of the column's settings. The most common reason for this is that the default value does not conform to the column's data type. For example, if you enter a letter as the default value of a column whose data type is bit, you will encounter this error.

Error validating the formula for column.

The formula contains an error. Common errors include mismatched parentheses, misspelled column names, and using an operator on a column whose data type does not support that operator. Choose Yes to modify the formula. Choose No to leave it as is.

For more information about formulas, see the Microsoft SQL Server documentation.

Function argument count error.

A function in your query has the wrong number of arguments. Review you query syntax to locate the function.

For more information, see [Functions for Expressions](#).

Identity column '<0s>' in table '<1s>' must have a datatype of int, smallint, tinyint, decimal or numeric with scale of 0.

Only certain datatypes can be used for Identity columns. Columns that are Identity columns contain system-generated, sequential values that uniquely identify each row within a table.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

Identity seed must be a number containing <0d> digits or less.

You have entered an invalid value for the identity seed field. Enter an integer whose magnitude is less than the maximum indicated in the error message. If the data type of the column is tinyint, you must enter a positive number. For any other data type, you can enter a positive or negative number.

If you save the view encrypted, you will no longer be able to alter the view definition.

By encrypting a view, you help prevent all users (regardless of database privileges), including yourself, from ever seeing the view definition. Thus, you will not be able to alter the view definition. Choose OK to save the view in encrypted form. Choose Cancel to cancel the Save operation.

Ignoring illegal use of ALL.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Although the query will run, you should review your query syntax and correct the error.

For more information, see [About Creating Queries](#).

Ignoring ODBC syntax.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Illegal sequence use.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Illegal use of collation clause.

The SQL statement uses the collation clause illegally. Consult the Microsoft SQL Server documentation for the proper use of this clause.

Illegal use of expression list.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Illegal use of outer join operator.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Create an Outer Join in a Query](#).

Incomplete column list.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Incomplete parameters list.

A parameter is missing from a user-defined function that returns a table. Rewrite the statement to supply the missing argument.

Incomplete parameters or column list.

An argument is missing from an insert statement. Rewrite the insert statement to supply the missing argument.

Incomplete SET clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Incomplete VALUES list.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be <2d> bytes or less.

This action will enlarge the index that is used as a key for full-text indexing of this table beyond the maximum size for such an index. If you proceed with the operation, full-text indexing for this table will be disabled. Choose Yes to proceed with the operation and disable full-text indexing. Choose No to cancel the operation.

Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be single-column.

You are adding a column to the index that is used as a key for full-text indexing of this table. Proceeding with the operation will disable full-text indexing for this table. Choose Yes to proceed with the operation and disable full-text indexing. Choose No to cancel the operation.

Index '<0s>' is used to enforce the full-text key on table '<1s>' and must be unique.

This operation will allow duplicate values on the index that is used as a key for full-text indexing of this table. If you proceed with the operation, full-text indexing for this table will be disabled. Choose Yes to proceed with the operation and disable full-text indexing. Choose No to cancel the operation.

Index '<0s>' is used to enforce the full-text key on table '<1s>' and must not be null.

You are performing an operation that will allow null values within the column that is used as a key for full-text indexing of this table. Proceed with the operation to disable full-text indexing for this table or cancel the operation to keep full-text indexing.

Index already exists.

The database already contains an index with the name you supplied. Choose another name.

Index must have at least one column.

Your modification would result in an index without columns, which is illegal.
You can delete the index or modify it by adding a column to it.

Indexed View must contain a clustered index.

Because an indexed view is stored on disk, it must contain a clustered index (rather than a nonclustered one). Modify the index to make it a clustered index.

Internal Error.

An unexpected error has occurred. You can try any of the following remedial actions:

- Try the operation again.
- Restart your application and try again.
- Restart your computer and try again.
- Check the Microsoft Knowledge Base, available from the Microsoft home page, for any available new information on internal errors.

Invalid data type.

The parameter's data type is not recognized by the database as a valid data type for function or procedure parameters. For a list of the value data types for function and procedure parameters, see the Microsoft SQL Server documentation.

Invalid entries must be resolved before you can exit this pane.

Exiting a pane causes other panes to be updated. If there is an invalid entry in the pane you are editing, you need to change or delete it.

Invalid escape character.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

Your escape character must be a single character.

For more information, see [About Creating Queries](#).

Invalid fill factor; enter an integer between 0 and 100.

Because fill factor must be a percentage, its minimum value is zero and its maximum value is 100. You can use only whole-number percentages. For example, 67 is acceptable, but 66.7 is not.

Invalid identifier '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Invalid name. You must provide a name for this object.

An object name is required to continue. Valid names begin with an underscore (_) or a character and contain a combination of characters, numbers, and underscores.

The maximum length for a name based on the server is 32 for Microsoft SQL Server 6.5.

In general, avoid names that contain periods.

Invalid or missing expression near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Invalid or missing expression.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Invalid prefix or suffix characters.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Invalid row index: Goto failed.

You have entered a non-numeric value into the GoTo record field. Enter a numeric value instead. Entering a number that is greater than the number of rows in the table, goes to the last record.

Invalid text or symbol.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Invalid view name.

The View name you've specified is not valid. Valid names begin with an underscore (_) or a character and contain a combination of characters, numbers and underscores.

The maximum length for View names is 32 for Microsoft SQL Server 6.5 and 128 for SQL Server 7.0. Do not use names with a period.

IS operator can only be used with NULL or NOT NULL.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Linked Server object cannot be used as a destination with this query type.

The selected query type does not support using linked server objects as destinations.

For more information, see [About Creating Queries](#).

Missing escape character in LIKE predicate.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing FROM clause near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing FROM clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing FROM keyword near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing FROM keyword.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing INTO keyword near '<0s>'.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing INTO keyword.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing or incomplete SELECT clause.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing pattern in LIKE predicate.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing SET keyword.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing subquery correlation name.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Missing subquery or the operator you entered requires parenthesis.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Create a Subquery](#).

Missing subquery.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Create a Subquery](#).

Must enter either TRUE or FALSE.

When specifying a criteria for the EXISTS statement, you must use either True for EXISTS or False for NOT EXISTS.

For more information, see [About Creating Queries](#).

Need alias for this field.

You cannot remove the alias from this output column. Reenter the alias and proceed.

NOT cannot be used in a column cell.

To negate a criterion that contains a subquery, you cannot place the NOT operator in the Column cell of the Grid pane. Instead, you should put the keyword FALSE in the Criteria cell of that same row of the Grid pane.

Object name cannot begin with #.

The object name cannot begin with a pound sign (#). Choose a different name.

Microsoft SQL Server uses a leading pound sign in an object name to indicate that the object is a temporary object.

Object with name already exists in the database.

The object you are trying to save has the same name as an object already in the database. Choose another name and proceed.

ODBC driver returned an invalid ODBC version and needs to be updated.

Your ODBC driver is not current. Go to the Microsoft home page and look for the latest version of the driver.

For more information refer to your ODBC documentation.

One or more selected tables are already on the database diagram.

You can only have one copy of any table on a database diagram. If you want to create a new table that has some of the same columns as an existing table, you can duplicate an existing table as the first step in creating a new table.

For more information, see [Duplicate a Table](#).

Only one clustered index can be created on table '<0s>'.

A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore only one clustered index can be created on each table.

To create a second index on the same table, change the Clustered property setting before creating the second index.

For more information, see [About Indexes](#) and [Create a Clustered Index](#).

Only one ROWGUID column is allowed per table.

You have attempted to assign more than one ROWGUID. The ROWGUID is a special property similar to IDENTITY and only one ROWGUID is allowed per table.

For more information, see [Set Column Properties](#).

OpenRowset cannot be used as a destination with this query type.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

ORDER BY not supported for CREATE TABLE query.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Reorder Output Columns in a Query](#).

Outer join column may not be used with an IN predicate or subquery.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Create Outer Joins in a Query](#).

Poorly formed comment.

The comment contains an error. Be sure that the body of the comment does not contain any comment markers (`/*` or `*/`).

Poorly formed Transact SQL CAST function.

An expression contains a bad Transact SQL CAST function. Consult the Microsoft SQL Server documentation for the syntax of this function.

Primary key or index cannot be created on a column with no name.

You have included in a primary key a column with no name. Provide a name for the column before including that column in the primary key.

Primary key or index cannot be created on column '<0s>' because its data type is '<1s>'.

You have included in an index or primary key a column whose data type precludes its participation in keys or indexes. Change the column's data type or remove it from the index or primary key.

For the rules restricting the data types of index and key columns, refer to the Microsoft SQL Server documentation.

Property cannot be modified.

You attempted to change a property that is incompatible with other properties already specified.

Query Designer Error.

No specific help is available for this error. If the text in the error message does not provide enough information for you to solve the problem, please consider the following options:

- Verify that your settings are correct and try the action again.
- Use the index to search for keywords related to the error message.
- If the problem is due to external applications or software, consult the documentation for that area. For example, errors may be caused by your database software, your server setup, or even the network you are using.
- For SQL statement errors, See the Microsoft SQL Server documentation to ensure you are using the appropriate syntax.
- Check the Readme for late-breaking information concerning the error.

Query Designer supports no more than 1 table for this type of query.

Only one data source can be used with this query type.

For more information, see [About Supported Query Types](#).

Query has executed successfully.

The query executed successfully.

Relationship '<0s>' was modified or deleted since the database diagram was loaded.

A modified relationship is about to be saved. It is possible that a table involved in the relationship has been deleted from the database by another user.

Verify whether this relationship is meant to be modified. Changing or deleting a relationship for a table may affect other tables.

For more information about relationships, see [About Table Relationships](#) and [Delete a Relationship](#).

Removing the formula from the selected column leaves it with data type that is disallowed in indexes and relationships.

The column whose formula you are removing participates in one or more indexes or relationships. The modification will result in a column whose data type precludes it from participating in them. Choose Yes to proceed with the modification and delete the indexes and relationships. Choose No to cancel the modification.

Row limit value should be between <xxx> and <xxx>.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Rules for Entering Search Values](#).

Rows pasted.

The rows you pasted in the results pane have been transmitted to the database. Because the database committed these modifications when you transmitted them, you cannot undo this change.

Select both primary key table and foreign key table before selecting any field for the relationship.

You tried to select the columns from one table of a relationship before indicating what the other table is. You must first indicate both tables before proceeding to choose columns.

**Setting for Length must be from <0d>
to <1d>.**

You must enter a whole number within the range specified in the error message.

**Setting for Precision must be from
<0d> to <1d>.**

You must enter a whole number within the range specified in the error message.

**Setting for Scale must be from <0d>
to <1d>.**

You must enter a whole number within the range specified in the error message.

SQL statement could not be parsed.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

SQL text cannot be represented in the grid pane and diagram pane.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

SQL Verify failed.

The database server could not verify the validity of the SQL statement. Be sure that the SQL syntax you are using is supported by the server.

Syntax error in table reference.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Syntax error in table reference: '<0s>' not recognized.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Table '<0s>' already exists.

The table name you specified already exists in the database. Type another name for the new table.

Table '<0s>' has no columns.

You tried to save a table with no columns. Either add a column to the table, delete the table from the database diagram, or close the Table Designer without saving the table.

Table '<0s>' is marked for deletion and was not added to the diagram.

A table that has been removed from the database diagram will be deleted from the database diagram when the changes are saved. The table will not, however, be deleted from the database.

For more information, see [Remove a Table from a Database Diagram](#).

-or-

The table has been deleted from the database by another user and cannot be added to your database diagram.

For more information, see [Delete a Table from a Database Diagram and the Database](#).

Table '<0s>' no longer exists in the database.

The table has been deleted from the database and cannot be used in your database diagram.

For more information, see [Delete a Table from a Database Diagram and the Database](#).

Table is not in the query definition.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Table name cannot include the double-quote character (").

The table name cannot include the double-quote character ("). Choose a different name.

The alias name is too long.

The name you supplied as an alias for a table, view, or function is too long. Enter a shorter name.

For information about the length restrictions and other rules for aliases, check the Microsoft SQL Server documentation.

The Allow Nulls property can't be set on a column that is part of the primary key.

All columns that are part of a table's a primary key must contain aggregate unique values other than NULL. To add the column to a primary key, first clear the Allow Nulls check box on the **Column Property** dialog box.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

The Allow Nulls property can't be set on column '<0s>' because it is an identity column.

All identity columns must contain unique values other than NULL. To make the column an identity column, first clear the Allow Nulls check box on the **Column Property** dialog box.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

The collation properties of columns <0s> and <1s> do not match.

Each pair of corresponding columns from the primary-key table and foreign-key table of a relationship must have identical collation settings, but this relationship includes a pair of corresponding columns with different settings. Remove the columns from the relationship or change the collation settings to be identical.

The columns in table '<0s>' do not match an existing enabled primary key or UNIQUE constraint.

At least one table participating in a relationship must have either a Primary Key or a Unique Constraint. After setting up a Primary Key or a Unique constraint for one of the tables you've selected, you can then define other relationships for that table.

For more information, see [Create a Unique Constraint](#) and [Define a Primary Key Constraint](#).

The columns in table '<0s>' do not match an existing primary key or UNIQUE constraint.

The columns on the primary key side of a foreign key relationship must participate in either a primary key or a unique constraint. After setting up a primary key or a unique constraint for one of the tables you've selected, you can then define other relationships for that table.

For more information, see [Create a Unique Constraint](#) and [Define a Primary Key Constraint](#).

The database row you are modifying no longer exists in the database.

After you retrieved data, another user deleted a row from the database. You are now trying to modify that row. Your modification will not work, because the row that is the target of your modification no longer exists.

The datatype of column '<1s>' in the '<0s>' table can't be changed because it participates in index '<2s>'.

The sum of the lengths of all columns in an index can not exceed 256 bytes. Changing to the specified datatype would exceed the maximum index size of <3d> bytes.

For more information, see [About Database Designer Considerations for SQL Server Databases](#) and [Create an Index](#).

The Default Value property can't be set on column '<0s>' because it is an identity column.

Identity columns contain system-generated sequential values that uniquely identify each row within the table. Therefore, identity columns cannot have default values.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

The existing relationship must have at least one pair of related columns.

There must be matching columns on both sides of a foreign key relationship. Deleting a column that is part of a relationship will disable the relationship.

For more information, see [Create a Relationship Between Tables](#).

The following datatype or size property of '<0s>.<1s>' doesn't match '<2s>.<3s>'.

When creating a foreign key relationship, the data types and other properties of selected columns must match. Select a column with the same data types and other properties as the related column. The properties that must match are Data type, Length, Precision, Scale, and Collation.

The following error was encountered while changing the collation: <0s>.

The collation was not changed because of the underlying problem described in the message. Changing a column's collation is akin to changing its data type. That is, after you change the collation, Microsoft Access validates the column's data type. This validation can uncover a number of problems, including:

- The modification enlarges the column, which in turn enlarges an index beyond the maximum size for indexes.
- The modification is incompatible with existing data values in the column.

Choose a different collation for the column or fix the underlying problem.

The following schema-bound objects will be modified.

The object you are modifying contributes to the definition of one or more schema-bound objects. Those schema-bound objects will be modified or dropped. The body of the message gives the details.

The identity increment must be a number containing <0d> digits or less.

You have entered an invalid value for the identity increment field. Enter an integer whose magnitude is less than the maximum indicated in the error message. If the data type of the column is tinyint, you must enter a positive number. For any other data type, you can enter a positive or negative number.

The Identity property can't be set on column '<0s>' because it allows null values.

Identity columns contain system-generated sequential values that uniquely identify each row within the table. Therefore, identity columns cannot have null values.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

The Identity property can't be set on column '<0s>' because it has a default value.

Identity columns contain system-generated sequential values that uniquely identify each row within the table. Therefore, identity columns cannot have default values.

For more information, see [About Database Designer Considerations for SQL Server Databases](#).

The index is used to enforce the full-text key for this table. Deleting this index will disable full-text indexing for the table. Do you want to proceed?

You are performing an operation that will delete an index that is used as a key for full-text indexing of this table. If you proceed with the operation, full-text indexing for this table will be disabled.

The name is too long.

The name supplied for a database object is too long. Enter a shorter name.

For information about the length restrictions and other rules for object names, check the Microsoft SQL Server documentation.

The new relationship must have at least one pair of related columns.

The relationship must relate at least one column from the foreign-key table to at least one column in the primary key table. Select a column from each table or delete the relationship.

The NOT keyword may not be used in a column cell.

When entering a NOT EXIST subquery into the Column cell, enter just the EXIST clause. Then go to the Criteria cell and enter False for it.

For more information, see [About Creating Queries](#).

The number of selected columns exceeds the number of columns in the target table.

You cannot create a compound foreign key relationship using more columns than exist in the target table. Make sure the number of columns you've selected matches the number in the target table.

For more information, see [Create a Relationship Between Tables](#).

The primary key or UNIQUE constraint cannot be changed until its existing relationships are deleted.

You cannot change a table's primary key or unique constraint while other foreign key relationships are still in place. To change a table's primary key or unique constraint, first delete all existing relationships for the table.

For more information, see [Delete a Relationship](#).

The primary key or Unique constraint cannot be changed while relationships to the existing primary key or Unique constraint are enforced.

You cannot change a table's primary key or unique constraint while other foreign key relationships are still in place. To change a table's primary key or unique constraint, first delete all existing relationships for the table.

For more information, see [Delete a Relationship](#).

The Query Designer does not support the <0s> SQL construct.

The syntax you entered is valid but is not supported visually by the Query Designer. Be sure to verify your syntax before saving.

For more information, see [About Supported Query Types](#).

The Query Designer does not support the critical ODBC APIs.

The API you are attempting to use is not supported. Refer to your ODBC driver documentation. Use Microsoft provided ODBC drivers whenever possible.

The Query Designer does not support the current ODBC API.

The API you are attempting to use is not supported. Refer to your ODBC driver documentation. Use Microsoft provided ODBC drivers whenever possible.

The specified OR group number is already in use.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

The SQL Server does not meet minimum requirements of version 6.5 service pack 3.

You are connected to a version of Microsoft SQL Server database that is older than version 6.5, service pack 3. Microsoft Access cannot work with that version of SQL Server.

Connect to a SQL Server database that is no older than SQL Server 6.5, service pack 3.

The SQL syntax has been verified against the data source.

The SQL syntax verified successfully. For more information about SQL syntax, see the Microsoft SQL Server documentation.

The table being loaded into memory has a user-defined data type that is not recognized.

The table you are loading into memory refers to a recently created user-defined data type, but your in-memory list of user-defined data types does not include it.

Because the data type was created recently, it is not present in local memory. All user-defined data types are loaded from the database into local memory when you open the first Table Designer window or database diagram within your connection to that database. To refresh your in-memory copy of the database's user-defined data types, close all open database diagrams and Table Designer windows within that database connection. After you close them all, reopen them in turn. When you reopen the first one, the list of user-defined data types will be refreshed.

For more information, see [About Multiuser Environments](#).

The table must have at least one column that is not computed.

You cannot create a table all of whose columns are computed. You can, however, create a view containing entirely computed data.

The table-valued function used as target is not an in-line function.

The operation you are attempting requires an updateable user-defined function (an in-line function), but the function you are using is not updateable. You can replace the function with an updateable one, or you can change the query type to a SELECT query.

The total size of an index or primary key cannot exceed 256 bytes.

The sum of the lengths of all columns in an index can not exceed 256 bytes. Changing to the specified datatype would exceed the maximum index size of <3d> bytes.

For more information, see [About Indexes](#), [About Database Designer Considerations for SQL Server Databases](#), and [Create an Index](#).

The value for the extended property is too long.

The database enforces a maximum length for the property, but the value you supplied exceeds this length. Use a shorter value and save again.

The value you entered is not consistent with the data type or length of the column.

Entered values must match the column data types.

For more information, see [Rules for Entering Search Values](#).

There are no columns selected. Please select one or more columns and re-run the query

The SELECT list does not contain any output columns. Specify the columns you want to be returned and run the query again.

For more information on SELECT statements, see the Microsoft SQL Server documentation.

There are not enough columns to match the value list.

An UPDATE or INSERT statement contains more values than the table being updated. Verify your SQL statement to make sure the number of columns matches the table.

There are not enough values in the subquery select list.

The SELECT statement needs to include the same number of columns that the embedded subquery returns.

For more information, see [Create a Subquery](#).

There are too many values from the sub-query select list.

The number of columns in the target of the INSERT operation exceeds the number of data values you supplied. Modify the query to include one value for each column.

There is no unique table in this query.

The query needs to unambiguously identify which table to return values from.

This backend version is not supported to design database diagrams or tables.

The database to which you are connected does not allow schema modification with Microsoft Access. Be sure you are connected to the correct database.

This cell contains the text string "<NULL>" which may not be processed correctly.

Use NULL or NOT NULL for comparing null data. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

This operator cannot be used with columns with data type "uniqueidentifier".

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

This SQL statement type cannot be used in a view.

A view definition must be a SELECT statement (rather than an INSERT, UPDATE, DELETE, or CREATE TABLE statement). Alter the view definition accordingly.

Too many characters for field width.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Too many or too few parameters specified.

You are using a user-defined function, but you are not supplying the correct number of parameters. Check the definition of the user-defined function to determine how many parameters it takes.

Unable to execute create script because it does not begin with CREATE or ALTER.

You are connected to a Microsoft SQL Server 7.0 or SQL Server 2000 database, which requires that create scripts begin with a CREATE statement or an ALTER statement.

Modify the script accordingly and run it again.

Unable to execute create script because it does not begin with CREATE.

You are connected to a Microsoft SQL Server 6.5 database, which requires that create scripts begin with a CREATE statement.

Modify the script accordingly and run it again.

Unable to execute the create script; the table name is invalid or missing.

You are attempting to create an object such as a trigger that is owned by a particular table or view, but the script text does not refer to that owning table or view. Modify the script to refer to the owning table or view and rerun the script.

For details about the syntax of scripts, refer to the Microsoft SQL Server documentation.

Unable to locate data source.

The specified data source cannot be found. Check the data source location and the entered location.

Unable to parse expression.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Unable to parse query text.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Unable to parse statement.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Unable to read object text; the text is encrypted.

You are trying to read the text of an object whose text is encrypted. Because the object's creator chose to encrypt the text, you will not be able to read it.

Unable to save; the object name is invalid or missing.

Your script omits the object name or uses an object name that does not conform to the naming rules of the database. Supply a valid name and save it again. For more information about naming rules for objects, see the Microsoft SQL Server documentation.

Unexpected object type.

Within your script, the CREATE statement is followed by an unrecognized word. The word following CREATE must be an object type recognized by the SQL syntax of your database. Make sure that you have not misspelled the word.

Unknown column.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Unknown conversion specification.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Unquoted alias contains white space.

An alias you supplied contains embedded blanks or other white space. Remove the white space or put double-quotation marks around the alias.

Use of GROUP BY function in WHERE clause not allowed.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [Summarize or Aggregate Values for All Rows in a Table](#).

Use of Transact SQL CONVERT function might be unnecessary.

There is an error in the syntax of your query. The Query Designer has attempted to locate the source of the error. Review your query syntax and correct the error before running your query.

For more information, see [About Creating Queries](#).

Values for '<0s>' must lie within the range <1d> to <2d>.

You entered a property value that is outside the range of legal values for that property. Choose a value within the range specified.

Values for property must lie within the range x to y.

The database enforces a minimum and maximum value for the property, but the value you supplied is outside that range. Modify the value accordingly and save again.

View already exists in the database.

Views must have unique names within the database. Enter a different name for the view.

View definition includes no output columns or no items in the FROM clause.

A view definition must have at least one table, view or function in the FROM clause, and must have at least one column in the select list. The view definition is missing one or both. Modify the view definition accordingly.

View has indexes. If you remove schema binding, the indexes will be dropped.

The operation you are performing will automatically remove schema binding from an indexed view. When schema binding is removed, the indexes will be dropped. Click OK to proceed with the operation; click Cancel to cancel the operation.

For more information, see [About Indexed Views and Schema Binding](#).

View name cannot begin with #.

The view name cannot begin with a pound sign (#). Choose a different name.

Warning: It is likely that the view definition will result in a view that cannot be indexed.

The view definition you are trying to transmit to the database might result in a view that cannot be indexed. You can proceed to save the view, but Microsoft SQL Server might not create the indexes for it.

Warning: It is likely that your modification will result in a view that cannot be indexed.

The modifications you are trying to transmit to the database might result in a view that cannot be indexed. If you proceed with the modifications, the view's existing indexes will be deleted.

WITH TIES clause requires an ORDER BY clause.

The query does not conform to SQL syntax, because it uses the WITH TIES clause without an attendant ORDER BY clause. Either remove the WITH TIES clause or add an ORDER BY clause.

You are about to delete a row or rows.

Choosing Yes will delete the selected row from the table. If you say Yes, the deletion will be immediately transmitted to the database. You will not be able to Undo this change.

You are about to paste n rows.

When you paste rows in the results pane, the rows are immediately transmitted to the database. Because the database commits these modifications when you transmit them, you cannot undo this change.

You are not logged in as the database owner or as a user that is a member of the db_owner role. You will not be able to save changes to tables that you do not own.

Because you are not logged on as the system administrator, database owner, or a user that is a member of the db_owner role, you have limited privileges to the database. The privileges you have are determined by the permissions granted to your logon ID and the privileges granted to the roles that your logon ID is a member of.

Even though you are not the database owner, you will still be able to use any tables that you have permissions to see. For example, you can create database diagrams using such tables. However you won't be able to perform all edits. Certain edits require SQL Server CREATE TABLE permission, which gives you permission to create new tables and modify tables that you own.

Even if you have CREATE TABLE permission, there are limitations to the modifications you can make. Remember, as you modify an existing table or design a new one, your work can induce attendant modifications in other tables. For example, if you change the data type of a foreign-key column, the corresponding column in the primary-key table will be automatically modified by Microsoft Access. If you do not own the primary-key table, and you are not logged in as the system administrator, database owner, or a user that is a member of the db_owner role, your modification will fail.

For more information, see [About Ownership of Database Objects](#).

You cannot create a view which is self-referenced.

The view definition's FROM clause includes the name of this view. Remove the view name from the FROM clause.

You entered <0d> characters for '<1s>'. The maximum number of characters allowed is <2d>.

You entered a property value that is longer than the maximum allowed for that property. Choose a shorter value.

You entered x characters for a property; the maximum allowed is y.

The database enforces a maximum length for the property, but the value you supplied exceeds this length. Use a shorter value and save again.

You might not have permission to perform this operation, or the object <0s> might no longer exist in the database.

The object you want to work with may be listed, but you may not have SELECT permission to this object. Or, the object may no longer exist in the database.

Refresh the list of objects to find out if this object still exists in the database. If this object still exists, then contact the owner of the object or the Database Administrator to get permission to this object.

For more information, see [About Ownership of Database Objects](#).

Your database diagram will be updated with the following changes to match the database before the following tables can be loaded.

A table you are trying to add to the database diagram is related to a table already on the database diagram, and that relationship somehow contradicts the contents of your database diagram. This occurs because another user has added or modified the relationship since you began working on the database diagram.

Remember, when you add a table to a database diagram, the Database Designer automatically includes any relationships between the existing tables and the newly added table. In attempting to add such a relationship, the Database Designer has discovered that your in-memory copy of the database structure no longer matches the contents of the database. Before adding the table, the Database Designer will update your database diagram accordingly.

For example, suppose your database diagram is initially consistent with the database. The database diagram includes two tables, `Student` and `Course`, and a one-to-one relationship between them, called `TutoringAssignment`. The relationship indicates that each student can tutor a course and each course can be tutored by a student. Initially, your diagram includes no other tables.

While you are working on the database diagram, another user changes the database structure. That user deletes the `TutoringAssignment` relationship and adds another relationship of the same name between the tables `Student` and `SectionOfCourse`. (This change improves the database structure, because it more faithfully represents your organization's information needs. Students are assigned to tutor particular sections of courses, not courses in general.) The other user commits these changes to the database.

Now you add the `SectionOfCourse` table to your database diagram. Because this table has a relationship to the `Student` table already present on your database

diagram, the Database Designer tries to include this relationship on the database diagram. The conflict arises because the to-be-included relationship has the same name as a now obsolete relationship already present on your database diagram.

Because of these conflicts, the Database Designer will update your in-memory copy of the database structure accordingly. That is, it will remove the TutoringAssignment relationship between the tables Student and Course. Only then will the Database Designer add to your database diagram the SectionOfCourse table and the TutoringAssignment relationship between the Student and SectionOfCourse tables.

Note The Database Designer does not reconcile all differences between your database diagram and the database. It reconciles only those differences that would otherwise prevent you from adding the table to the database diagram.

Your entry cannot be converted to a valid date time value.

In general, the ANSI standard date format is used with databases that represent dates using a true date data type. In contrast, the datetime format is used with databases that support a datetime data type.

For more information, see [Rules for Entering Search Values](#).

Your Logon does not have CREATE TABLE permission; you might not be able to make certain edits that require this permission.

You are creating or modifying a table in a database in which your logon ID does not have CREATE TABLE permissions and is not a member of the **db_owner** role. Check with the database administrator to make sure that you have the necessary permissions to create tables in the database.

Remember, a seemingly innocuous modification to an existing table can require CREATE TABLE permission. For example, if you add a column to a table, Microsoft Access instructs Microsoft SQL Server to delete the table, re-create it, and re-insert the rows it had contained.

Your table will be updated with the following changes to match the database.

The Table Designer needs to load information about the table to which you are trying to create a relationship. For example, it needs to load the details about the other table's columns, so you can choose which columns from the table you are designing correspond to columns in the other table.

The contents of the database and the in-memory contents of the Table Designer are in conflict. This conflict occurs because another user has recently modified the table you are working on, but the Table Designer retains the older version of the table in your local memory.

The body of the message shows the details of the conflict. Before proceeding to load the other table into your local memory, the Table Designer will reconcile these conflicts accordingly.

Note The Table Designer does not reconcile all differences between your local memory and the database. It reconciles only those differences that would otherwise prevent you from creating the relationship you are trying to create.

[Show All](#)

Create an indexed view (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

An indexed view is a view whose result set is stored in the database for fast access. Indexed views are supported when your [Microsoft Access project](#) is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

1. In the Database window, click Queries  under **Objects**, click the query you want to open, and then click **Design** on the database window toolbar.
2. Add tables to the Diagram pane. Because you intend this to be an indexed view, be sure to add only tables that you own.
3. Select which columns you want included in the view. Do not use the asterisk (*); you must explicitly select each column that you want to appear in the indexed view.
4. Right-click the background of the Diagram pane, then choose **Manage Indexes** from the shortcut menu. The **Indexes** dialog box appears.

Note There are many situations in which the **Manage Indexes** command is unavailable, because there are many restrictions on which views can be indexed. For example, you cannot index a view unless you are the owner of each table contributing to it. For a complete description of the restrictions on creating indexed views, see the Microsoft SQL Server documentation.

5. Within the dialog box, click **New**.
6. Supply the information for the index definition— index name, index columns and their order, index file group, and the other index settings. For more information on each dialog box option, click **Help**.

[Show All](#)

Create a clustered index (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Background information](#)

In Microsoft SQL Server databases you can create a clustered index. In a clustered index, the physical order of the rows in the table is the same as the logical (indexed) order of the index key values. A table can contain only one clustered index. UPDATE and DELETE operations are often accelerated by clustered indexes because these operations require large amounts of data to be read. Creating or modifying a clustered index can be time-consuming, because it is during these operations that the table's rows are reorganized on disk.

Consider using a clustered index for:

- Columns that contain a limited number of unique values, such as a state column that contains only 50 unique state codes.
- Queries that return a range of values, using operators such as BETWEEN, >, >=, <, and <=.
- Queries that return large result sets.

[Create a clustered index](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table you want to index, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose the **Indexes/Keys** tab, and then create a new index. For more information on each dialog box option, click **Help**.

To modify an existing index, select the index from the list.

4. Select the **Create as CLUSTERED** check box.

The index is created in the database when you save the database diagram.

Attach a new check constraint to a table or column (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Attach a check constraint to a table to specify the data values that are acceptable in one or more columns.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table that will contain the constraint, then select **Constraints** from the shortcut menu.
3. Choose **New**. The **Selected constraint** box displays the system-assigned name of the new constraint. System-assigned names begin with "CK_" followed by the table name.
4. In the **Constraint expression** box, type the SQL expressions for the check constraint. For example, to limit the entries in the state column of the authors table to New York, type:

```
state = 'NY'
```

Or, to require entries in the zip column to be 5 digits, type:

```
zip LIKE '[0-9][0-9][0-9][0-9][0-9]'
```

Note Make sure to enclose any non-numeric constraint values in single quotation marks (').

5. If you want to give the constraint a different name, type the name in the **Constraint name** box.
6. Use the check boxes to control when the constraint is enforced:
 - To test the constraint on existing data before creating the constraint, check **Check existing data on creation**.

- To enforce the constraint whenever a replication operation occurs on this table, check **Enforce constraint for replication**.
- To enforce the constraint whenever a row of this table is inserted or updated, check **Enforce constraint for INSERTs and UPDATEs**.

Delete a check constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Delete a check constraint when you want to remove the limitations on data values that are accepted in the column or columns included in the constraint expression.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Constraints** from the shortcut menu.
3. Select the constraint from the **Selected constraint** list.
4. Choose **Delete**.

Caution Choosing **Delete** will result in an action that cannot be undone without losing all other changes made to the database diagram. To undo this action, close this database diagram and all other open database diagrams without saving the changes.

The constraint is deleted from the database when you save the database diagram.

Rename a check constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table containing the constraint, then select **Constraints** from the shortcut menu.
3. Select the constraint you want to change from the **Selected constraint** list.
4. Type a new name in the **Constraint name** box.

The constraint name is updated in the database when you save your database diagram.

[Show All](#)

About check constraint expressions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [Overview of check constraint expressions](#)

When you attach a check constraint to a table or column, you must include an SQL expression.

You can create a simple constraint expression to check data for a simple condition; or you can create a complex expression, using Boolean operators, to check data for several conditions. For example, suppose the authors table has a zip column where a 5-digit character string is required. This sample constraint expression guarantees that only 5-digit numbers are allowed:

```
zip LIKE '[0-9][0-9][0-9][0-9][0-9]'
```

Or suppose the sales table has a column called qty which requires a value greater than 0. This sample constraint guarantees that only positive values are allowed:

```
qty > 0
```

Or suppose the orders table limits the type of credit cards accepted for all credit card orders. This sample constraint guarantees that if the order is placed on a credit card, then only Visa, MasterCard, or American Express is accepted:

```
NOT (payment_method = 'credit card') OR  
(card_type IN ('VISA', 'MASTERCARD', 'AMERICAN EXPRESS'))
```

 [Defining a constraint expression](#)

In the **Check Constraints** tab of the property page, type an expression in the **Constraint expression** box using the following syntax:

```
{constant | column_name | function | (subquery)}  
[ {operator | AND | OR | NOT}  
{constant | column_name | function | (subquery)}...]
```

The SQL syntax is made up of the following parameters:

Parameter	Description
<i>constant</i>	A literal value, such as numeric or character data. Character data must be enclosed within single quotation marks (' ').
<i>column_name</i>	Specifies a column.
<i>function</i>	A built-in function. For details about functions, see the SQL Server documentation.
<i>operator</i>	An arithmetic, bitwise, comparison, or string operator. For details about operators, see the SQL Server documentation. Use in Boolean expressions to connect two expressions. Results are returned when both expressions are true.
AND	When AND and OR are both used in a statement, AND is processed first. You can change the order of execution by using parentheses. Use in Boolean expressions to connect two or more conditions. Results are returned when either condition is true.
OR	When AND and OR are both used in a statement, OR is evaluated after AND. You can change the order of execution by using parentheses. Negates any Boolean expression (which can include keywords, such as LIKE, NULL, BETWEEN, IN, and EXISTS).
NOT	When more than one logical operator is used in a statement, NOT is processed first. You can change the order of execution by using parentheses.

Types of joins (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

When you join tables, the type of join that you create affects the rows that appear in the result set. You can create the following types of joins:

- **Inner join** A join that displays only the rows that have a match in both joined tables. (This is the default type of join in the Query Designer.) For example, you can join the `titles` and `publishers` tables to create a result set that shows the publisher name for each title. In an inner join, titles for which you do not have publisher information are not included in the result set, nor are publishers with no titles. The resulting SQL for such a join might look like this:

```
SELECT title, pub_name
FROM   titles INNER JOIN publishers
       ON titles.pub_id = publishers.pub_id
```

Note Columns containing NULL do not match any values when you are creating an inner join and are therefore excluded from the result set. Null values do not match other null values.

- **Outer join** A join that includes rows even if they do not have related rows in the joined table. You can create three variations of an outer join to specify the unmatched rows to be included:
 - **Left outer join** All rows from the first-named table (the "left" table, which appears leftmost in the JOIN clause) are included. Unmatched rows in the right table do not appear. For example, the following SQL statement illustrates a left outer join between the `titles` and `publishers` tables to include all titles, even those you do not have publisher information for:

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
FROM titles LEFT OUTER JOIN publishers
```

```
ON titles.pub_id
= publishers.pub_id
```

- **Right outer join** All rows in the second-named table (the "right" table, which appears rightmost in the JOIN clause) are included. Unmatched rows in the left table are not included. For example, a right outer join between the `titles` and `publishers` tables will include all publishers, even those who have no titles in the `titles` table. The resulting SQL might look like this:

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
FROM titles RIGHT OUTER JOIN publishers
      ON titles.pub_id
      = publishers.pub_id
```

- **Full outer join** All rows in all joined tables are included, whether they are matched or not. For example, a full outer join between `titles` and `publishers` shows all titles and all publishers, even those that have no match in the other table.

```
SELECT titles.title_id,
       titles.title,
       publishers.pub_name
FROM titles FULL OUTER JOIN publishers
      ON titles.pub_id
      = publishers.pub_id
```

- **Cross join** A join whose result set includes one row for each possible pairing of rows from the two tables. For example, `authors CROSS JOIN publishers` yields a result set with one row for each possible author/publisher combination. The resulting SQL might look like this:

```
SELECT *
FROM authors CROSS JOIN publishers
```

Modify a primary key constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Modify a primary key if you want to change the column order, index name, clustered option, or fill factor.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table whose primary key you want to modify, and choose **Indexes/Keys** from the shortcut menu.
3. Select the primary key index from the **Selected index** list.
4. Type a new name in the **Index name** box. Make sure that your new name does not duplicate a name in the **Selected index** list.

The primary key is updated in the database when you save your database diagram.

Modify a foreign key constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Modify the foreign key side of a relationship if you want to change which columns are related to columns in the primary key table.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the relationship corresponding to the foreign key you want to modify, and then choose **Properties** from the shortcut menu.
3. Choose the **Relationships** tab.
4. Select the relationship from the **Selected relationship** list.
5. In the **Foreign key table** column, expand the list in the first row.
6. Select a different table column from the list. The foreign key column must match the data type and size of the primary key column, with these exceptions:
 - A char column or sysname column can relate to a varchar column.
 - A binary column can relate to a varbinary column.
 - A user-defined data type can relate to its base type.

Any changes you make to the relationship's properties take effect as soon as you move outside the grid in the property pages. The constraint is updated in the database when you save your database diagram.

Note Modifying a relationship in a database diagram marks both of the related tables as modified. Consequently, each table will also be marked as modified in any other database diagrams in which it appears.

Create a unique constraint (ADP)

Create a unique constraint to ensure no duplicate values are entered in specific columns that do not participate in a primary key. While both a unique constraint and a primary key enforce uniqueness, you should attach a unique constraint instead of a primary key constraint to a table if:

- You want to enforce uniqueness in a column or combination of columns. You can attach multiple unique constraints to a table, whereas you can attach only one primary key constraint to a table.
- You want to enforce uniqueness in a column that allows null values. You can attach unique constraints to columns that allow null values, whereas you can attach primary key constraints only to columns that do not allow null values. When you attach a unique constraint to a column allowing null values, you ensure that at most one row will have a null value in the constrained column.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, right-click the table that will contain the constraint, then select **Indexes/Keys** from the shortcut menu.
3. Choose **New**. A system-assigned name appears in the **Index name** box.
4. Under **Column name**, expand the list of columns and select the column that you want to attach the constraint to. To attach the constraint to multiple columns, select the additional columns in subsequent rows.
5. Select the **Create UNIQUE** check box.
6. Select the **Constraint** option.

The unique constraint is created in the database when you save the database diagram.

You can control the sort order of key values and the action taken when duplicate keys exist. To do this, you should create a unique index instead of a unique constraint.

[Show All](#)

Create a unique index (ADP)

[Background information](#)

In SQL Server, you can create a unique index when uniqueness is a characteristic of the data itself, but the combination of indexed columns is not the same as the table's primary key. For example, if you plan to query frequently on the Social Security number (ssn) column in the employee table (where the primary key is emp_id), and you want to ensure Social Security numbers are unique, create a unique index on ssn. If the user enters the same Social Security number for more than one employee, the database displays an error and cannot save the table.

When you create or modify a unique index, you can set an option to ignore duplicate keys. If this option is set and you attempt to create duplicate keys by adding or updating data that affects multiple rows (with the INSERT or UPDATE statement), the row that causes the duplicates is not added or, in the case of an update, discarded.

For example, if you try to update "Smith" to "Jones" in a table where "Jones" already exists, you end up with one "Jones" and no "Smith" in the resulting table. The original "Smith" row is lost because an UPDATE statement is actually a DELETE followed by an INSERT. "Smith" was deleted and the attempt to insert an additional "Jones" failed. The whole transaction cannot be rolled back because the purpose of this option is to allow a transaction in spite of the presence of duplicates.

[Create a unique index](#)

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the database window toolbar.
2. In your database diagram, select the table you want to index, right-click the table, and choose **Properties** from the shortcut menu.
3. Choose **Indexes/Keys**. The **Selected index** list displays the system-assigned name of the new index.
4. Under **Column name**, select the columns you want to index. You can select

up to 16 columns. For optimal performance, select only one or two columns per index. For each column you select, indicate whether the index arranges values of this column in ascending or descending order.

5. Select the **Create UNIQUE** check box.
6. Select the **Index** option.
7. Select the **Ignore duplicate key** option if you want to ignore new or updated data that would create a duplicate key in the index (with the INSERT or UPDATE statement).

The index is created in the database when you save the database diagram.

Note You cannot create a unique index on a single column if that column contains NULL in more than one row. Similarly, you cannot create a unique index on multiple columns if the combination of columns contains NULL in more than one row. These are treated as duplicate values for indexing purposes

Define a primary key constraint (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Define a primary key to enforce uniqueness for values entered in specified columns that do not allow nulls. If you define a primary key for a table in your database, you can relate that table to other tables, thus reducing the need for redundant data. A table can have only one primary key.

1. In the Database window, click **Database Diagrams**  under **Objects**, click the database diagram you want to open, and then click **Design** on the Database window toolbar.
2. In your database diagram, click the row selector for the database column you want to define as the primary key. If you want to select multiple columns, hold down the CTRL key while you click the row selectors for the other columns.
3. Right-click the row selector for the column and select **Primary Key**. A primary key index, named "PK_" followed by the table name, is automatically created; you can find it on the **Indexes/Keys** tab of the **Properties** page.

Warning If you want to redefine the primary key, any relationships to the existing primary key must be deleted before the new primary key can be created. A message will warn you that existing relationships will be automatically deleted as part of this process.

A primary key column is identified by a primary key symbol  in its row selector.

If a primary key consists of more than one column, duplicate values are allowed in one column, but each combination of values from all the columns in the primary key must be unique.

If you define a compound key, the order of columns in the primary key matches

the order of columns as shown in the table in your database diagram. However, you can change the order of columns after the primary key is created. In the **Column name** grid, remove the columns from the primary key. Then add the columns back in the order you want. To remove a column from the key, simply remove the column name from the Column name list.

[Show All](#)

About multiuser environments (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Overview of multiuser environments](#)

A multiuser environment is one in which other users can connect and make changes to the same database that you are working with. As a result, several users might be working with the same database objects at the same time. Thus, a multiuser environment introduces the possibility of your database diagrams being affected by changes made by other users, and vice versa. Such changes could include changes to copies of your database diagrams, other users' database diagrams that share database objects with your database diagrams, or the underlying database.

A key issue when working with databases in a multiuser environment is access permissions. The permissions you have for the database determine the extent of the work you can do with the database. For example, to make changes to objects in a database, you must have the appropriate write permissions for the database. For more information about permissions in your database, see the SQL Server documentation.

As one of multiple users, you may need to address any of the following issues when working with the Database Designer:

- Ownership of database objects
- Diagrams affected by another user's changes
- Database objects deleted by another user

[Ownership of database objects](#)

Each database object is owned by either a user or a role. If the owner is a role, the object is co-owned by every user belonging to that role.

Each object also has privileges associated with it. A privilege grants a particular authority to an object for either an individual user or for all users belonging to a particular role.

Depending on the privileges defined in the database, you can view objects that you do not own. For example, you can include on a database diagram an object owned by another user. When such an object appears on a database diagram or in the Table Designer, it is labeled with the owner name as well as the object name.



[Diagrams affected by another user's changes](#)

In a multiuser environment, your database diagram can be affected by changes other users have saved to:

- Your database diagram, which other users changed since you opened the diagram.
- Database objects shared between database diagrams.
- The database.

For example, your database diagram might contain a table that another user deleted or renamed. In such a case, your database diagram will no longer reflect the current state of the database. When you attempt to save your database diagram or selected tables, the **Database Changes Detected** dialog box notifies you that the database has been updated since you opened your diagram.

This dialog box also displays a list of database objects that will be affected as a result of saving your database diagram or selection. At this point, you can take one of these actions:

- **Save your database diagram or selection and update the database with all the changes in the list.** This action will also affect other database diagrams that share the same database objects.

For example, suppose you edit the `au_id` column in the `titleauthors` table on your database diagram and another user's database diagram contains the `authors` table which is related to the `titleauthors` table by the `au_id` column. Saving your database diagram will affect the other user's database diagram. Similarly, suppose that another user defined a check constraint for the `qty` column in the `sales` table. If you delete the `qty` column and save the `sales` table, the other user's check constraint will be affected.

- **Cancel the save action.** You can then close the database diagram without saving it. When you reopen the database diagram, it will be in synch with the database.
- **Save a list of the changes.** You can save the list of database changes shown in the **Database Changes Detected** dialog box to a text file so that you can investigate the cause of other users' changes. For example, if another user edited a table that you marked for deletion, you may want to research whether the table should be deleted before updating the database.

[Database objects deleted by another user](#)

In a multiuser environment, deleting database objects from the database can affect other users. If another user deletes a database object that appears in your database diagram, the effect of the deletion on your database diagram depends on where the object was deleted.

Where the object was deleted	How your database diagram is affected
In another copy of your database diagram	The deleted object will be removed from your database diagram next time you open it. If your database diagram is open when the deletion occurs, it still shows the deleted item.
In a different database diagram	The deleted object will be removed from your database diagram next time you open it. If your database diagram is open when the deletion occurs, it still shows the deleted item. If your database diagram has no unsaved changes to that object, the object will be removed from your database diagram the next time you open it.
In the database	If your database diagram contains unsaved changes to that object, the object will remain in your database diagram. You can recreate it in the database by saving your database diagram.

Note Recreating a deleted object creates a new definition of that object in the database; it does not restore the data that was deleted when the object was

deleted.

For example, if the `titles` table is deleted in the database and you have unsaved changes to the `titles` table in your database diagram, then saving your database diagram will create a new `titles` table in the database. Any data that existed in the `titles` table before it was deleted is not restored.

[Show All](#)

Structure of expressions (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

 [General guidelines for creating expressions](#)

An expression consists of any combination of column names, literals, operators, or functions. Follow these guidelines in combining elements to form expressions:

- Reference columns by typing their names. If your query uses more than one table and if you use a column name that is not unique, you must add the table name and a period to the column name. The following example shows the column name `job_id` qualified with the table name `employee`:

```
employee.job_id
```

- Include literal text by enclosing it in single quotation marks; no quotation marks are necessary for numbers.

Note The Query Designer can accept terms in double quotation marks, but might interpret them differently than you expect. In a Microsoft Access Project, the Query Designer always interprets double quotation marks as database object delimiters.

- Use standard arithmetic operators for numbers and a concatenation operator for combining strings.
- Include parentheses to establish precedence of operators.
- If you include a function, use these same guidelines for the arguments passed to the function. That is, reference columns by typing their names, enclose literal text in single quotation marks, and so on.
- If you pass column names as function arguments, be sure the data type of the column is appropriate for the function argument.
- You can include user-defined functions returning a scalar value in an expression.

Expression examples

The following table illustrates the use of expressions in a query.

Expression	Result
<pre>SELECT (price * .9) FROM products</pre>	Displays a discounted price (10% off the value in the price column).
<pre>SELECT (lname + ', ' + fname) FROM employee</pre>	Displays the concatenated values of the last name and first name columns with a comma between them.
<pre>SELECT sales.qty, titles.price FROM sales INNER JOIN titles ON sales.title_id = titles.title_id ORDER BY (sales.qty * titles.price)</pre>	After joining two tables, sorts the result set by the total value of an order (quantity times price).
<pre>SELECT au_lname, au_fname FROM authors WHERE (SUBSTRING(phone, 1, 3) = '415')</pre>	Displays authors whose area code is in the San Francisco area.
<pre>SELECT ord_num, ord_date FROM sales WHERE (ord_date >= DATEADD(day, -10, GETDATE()))</pre>	Finds all orders in the sales table that were made in the last 10 days. Today's date is returned by the GETDATE() function.

[Show All](#)

About indexed views and schema binding (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

[Indexed views and schema binding](#)

Indexed views are views whose results are persisted in the database and indexed for fast access. Indexed views are supported when your [Microsoft Access project](#) is connected to Microsoft SQL Server 2000 Enterprise Edition or SQL Server 2000 Developer Edition, but they are not supported in Microsoft SQL Server 2000 Desktop Edition. Indexed views are best used when data is read-only (such as a decision support system), queries of the indexed view do not involve aggregates or joins, and the base table schema definitions of the indexed view are not likely to change. For more information on indexed views, see the SQL Server documentation.

As with any other views, indexed views depend on base tables for their data. Such dependency means that if you change a base table contributing to an indexed view, the indexed view might become invalid. For example, renaming a column that contributes to a view invalidates the view. To prevent such problems, SQL Server supports creating views with “schema binding.” Schema binding prohibits any table or column modification that would invalidate the view. Any indexed view you create with the Query Designer automatically gets schema binding, because SQL Server requires that indexed views have schema binding. Schema binding does not mean you cannot modify the view; it means you cannot modify the underlying tables or views in ways that would change the view’s result set.

[What happens when you modify a base table or column](#)

Using the Table Designer or Database Designer, you might attempt to modify a base table or column that contributes to a view defined with schema binding. If your attempted modification could invalidate the view, the Designer warns you

and asks you whether you want to proceed. If you choose to proceed, these things happen:

- Your modifications to the base table occur.
- All views depending on the base table views are changed so that “schema binding” is removed. Thus, your subsequent changes to the base table will proceed without warning.
- If the dependent views were indexed, the indexes are deleted.

The Table Designer and Database Designer warn you before modifying a base table only if that base table contributes to a view with schema binding and if your modification does one or more of the following:

- Deletes the base table
- Renames the base table
- Recreates the base table
- Removes a column from the base table and the view includes that column
- Renames a column from the base table and the view includes that column

About ownership of database objects (ADP)

Note The information in this topic applies only to a Microsoft Access project (.adp).

Each database object is owned by either a user or a role. If the owner is a role, the object is co-owned by every user belonging to that role.

Each object also has privileges associated with it. A privilege grants a particular authority to an object for either an individual user or for all users belonging to a particular role.

Depending on the privileges defined in the database, you can view objects that you do not own. For example, you can include on a database diagram an object owned by another user. When such an object appears on a database diagram or in the Table Designer, it is labeled with the owner name as well as the object name.