

## Installing SQL Server

## Overview of Installing SQL Server 2000

Microsoft® SQL Server™ 2000 Setup creates a new installation of SQL Server 2000 or upgrades an earlier version. Before installing or upgrading to SQL Server 2000 it is recommended that you review the following topics.

Topic	Description
<a href="#">Preparing to Install SQL Server 2000</a>	Lists steps to take before running SQL Server 2000 Setup
<a href="#">SQL Server 2000: Editions and Components</a>	Presents an overview of the editions of SQL Server 2000, installation options, and components
<a href="#">Upgrading an Existing Installation of SQL Server</a>	Outlines options for upgrading to SQL Server 2000 from an earlier version

For more information, see related topics in the table below.

To install	See
SQL Server 2000 (typical installation of the relational database engine, client tools, and client connectivity components)	<a href="#">How to install SQL Server 2000 (Setup)</a>
SQL Server 2000 client tools only (includes management tools and client connectivity components, no server required)	<a href="#">How to install client tools only (Setup)</a>
SQL Server 2000 connectivity only (installs only the client connectivity components, no other options)	<a href="#">How to install connectivity only (Setup)</a>
A named instance or multiple instances of SQL Server 2000	<a href="#">How to install a named instance of SQL Server 2000 (Setup)</a>
A SQL Server 2000 virtual server for failover clustering	<a href="#">Before Installing Failover Clustering</a>

Analysis Services	<a href="#">Installing Analysis Services</a>
English Query	<a href="#">Installing English Query</a>

## Installing SQL Server

# Preparing to Install SQL Server 2000

Before installing Microsoft® SQL Server™ 2000, consider the following:

- Be sure the computer meets the system requirements for Microsoft SQL Server 2000. For more information, see [Hardware and Software Requirements for Installing SQL Server 2000](#).
- Back up your current installation of Microsoft SQL Server if installing SQL Server 2000 on the same computer.
- If installing a failover cluster, disable NetBIOS on all private network cards before running SQL Server Setup. For more information, see [Before Installing Failover Clustering](#).
- Review all SQL Server installation options and be prepared to make the appropriate selections when running Setup. For more information about editions and components, see [SQL Server 2000: Editions and Components](#).
- If you plan to install SQL Server to a location other than the default file locations, see [File Paths for SQL Server 2000](#).
- If using an operating system with Regional settings other than English (United States), or if customizing character set or sort order settings, review topics on collation settings. For more information, see [Collation Options for International Support](#).

## Before Running SQL Server 2000 Setup

Before running Setup:

- Create one or more domain user accounts if installing SQL Server 2000 on a computer running Microsoft Windows NT® or Microsoft

Windows® 2000, and you want SQL Server 2000 to communicate with other clients and servers. For more information, see [Creating Security Accounts](#).

- Log on to the operating system under a user account that has local administrative permissions, or assign the appropriate permissions to the domain user account.
- Shut down all services dependent on SQL Server. This includes any service using ODBC, such as Microsoft Internet Information Services (IIS).
- Shut down Microsoft Windows NT Event Viewer and registry viewers (Regedit.exe or Regedt32.exe).

## Installing SQL Server

# Hardware and Software Requirements for Installing SQL Server 2000

The minimum hardware and software requirements for running Microsoft® SQL Server™ 2000 are listed in the following tables.

## Hardware Requirements

This table shows hardware requirements for installing Microsoft SQL Server 2000 or SQL Server client management tools and libraries.

Hardware	Minimum requirements
Computer	Intel® or compatible Pentium 166 MHz or higher.
Memory (RAM) <sup>1</sup>	Enterprise Edition: 64 MB minimum, 128 MB or more recommended Standard Edition: 64 MB minimum Personal Edition: 64 MB minimum on Windows 2000, 32 MB minimum on all other operating systems Developer Edition: 64 MB minimum Desktop Engine: 64 MB minimum on Windows 2000, 32 MB minimum on all other operating systems
Hard disk space <sup>2</sup>	SQL Server database components: 95 to 270 MB, 250 MB typical Analysis Services: 50 MB minimum, 130 MB typical English Query: 80 MB Desktop Engine only: 44 MB

Monitor	VGA or higher resolution 800x600 or higher resolution required for the SQL Server graphical tools
Pointing device	Microsoft Mouse or compatible
CD-ROM drive	Required

1 Additional memory may be required, depending on operating system requirements.

2 Actual requirements will vary based on your system configuration and the applications and features you choose to install.

**Note** Microsoft SQL Server 2000 does not have a hardware compatibility list (HCL). If your computer meets the minimum requirements listed in the preceding table, SQL Server 2000 software works on the hardware certified for use with the Microsoft Windows® operating system. For more information about hardware certified for use with the Windows operating system, see the Microsoft Windows Hardware Compatibility List at [Microsoft Web site](#).

## Operating System Requirements

This table shows the operating systems that must be installed to use the various editions or components of Microsoft SQL Server 2000.

SQL Server edition or component	Operating system requirement
Enterprise Edition	Microsoft Windows NT Server 4.0, Microsoft Windows NT Server Enterprise Edition 4.0, Windows 2000 Server, Windows 2000 Advanced Server, and Windows 2000 Data Center Server.  Note that Microsoft Windows 2000 Server (any version) is required for some SQL Server 2000 features.
Standard Edition	Microsoft Windows NT Server 4.0, Windows 2000 Server, Microsoft Windows NT Server Enterprise Edition, Windows 2000 Advanced Server, and Windows 2000 Data Center Server.
Personal Edition	Microsoft Windows Me, Windows 98, Windows NT Workstation 4.0, Windows 2000 Professional, Microsoft Windows NT Server 4.0, Windows 2000

	Server, and all the more advanced Windows operating systems.
Developer Edition	Microsoft Windows NT Workstation 4.0, Windows 2000 Professional, and all other Windows NT and Windows 2000 operating systems.
Client Tools Only	Microsoft Windows NT 4.0, Windows 2000 (all versions), Windows Me, and Windows 98.
Connectivity Only	Microsoft Windows NT 4.0, Windows 2000 (all versions), Windows Me, Windows 98, and Windows 95.

**Note** Microsoft Windows NT® Server 4.0, Service Pack 5 (SP5) or later must be installed as a minimum requirement for all SQL Server 2000 editions.

SQL Server 2000 is not supported on Windows NT 4.0 Terminal Server.

For installations of SQL Server 2000 Personal Edition on Windows 98 computers without a network card, Windows 98 Second Edition is required.

## Internet Requirements

This table shows Internet requirements related to using Microsoft SQL Server 2000.

Component	Requirement
Internet software	<p>Microsoft Internet Explorer 5.0 is required for all installations of Microsoft SQL Server 2000, as it is required for Microsoft Management Console (MMC) and HTML Help. A minimal install is sufficient, and Internet Explorer is not required to be the default browser.</p> <p>Exception to the Internet Explorer 5.0 requirement: If using the Connectivity Only option and not connecting to a server that requires encryption, Microsoft Internet Explorer 4.01 with Service Pack 2 is sufficient.</p>

Internet Information Services	If writing XML applications, see <a href="#">System Requirements for the IIS Virtual Directory Management for SQL Server Utility</a> .
-------------------------------	--

## Network Software Requirements

Microsoft Windows NT, Windows 2000, Windows Me, Windows 98, and Windows 95 have built-in network software. Additional network software is required only if you are using Banyan VINES or AppleTalk ADSP. Novel NetWare IPX/SPX client support is provided by the NWLink protocol of Windows Networking.

**Note** TCP/IP must be enabled at the operating system level before installing SQL Server 2000. For more information, see [Network Libraries](#).

## Supported Clients

Microsoft SQL Server 2000 supports the following clients: Windows NT Workstation, Windows 2000 Professional, Windows 98, Windows 95, Apple Macintosh®, OS/2, and UNIX. Macintosh, OS/2, and UNIX do not support the SQL Server graphical tools and require ODBC client software from a third-party vendor.

## Considerations for Other Microsoft Products

The following Microsoft products require Service Release or Service Packs to operate correctly with SQL Server 2000.

### Access 2000

Microsoft Access 2000 requires the installation of either Microsoft Office 2000 Service Release 1 (SR1) or Access 2000 SR1 to operate correctly with SQL Server 2000. If running an earlier version of Access 2000, you cannot test automatic data processing (ADP) applications against SQL Server 2000. You cannot access database diagrams, stored procedures, table designs, or view designs.

Other issues to be addressed in a future Access Service Release:

- When you run Access 2000 with SR1, you can test ADP applications. You can also alter database diagrams, stored procedures, table designs, or view designs, but you cannot save any changes. A future Access Service Release will allow limited ability to save changes.
- The Access 2000 Create Database Wizard cannot successfully create a SQL Server 2000 database. You can work around this by first creating the database using SQL Server Enterprise Manager, and then creating an ADP for the database using the **Project (Existing Database)** option on the **New** dialog box in Access 2000.
- The Access 2000 Upsizing Wizard does not support upsizing to SQL Server 2000. You can work around this by using Data Transformation Services in the Enterprise Manager to import your MDB database file into SQL Server. You can then rename your MDB tables and create linked tables to the resulting SQL Server database with the same names as your original MDB table names.

## Visual Studio 6.0

When you run Microsoft Visual Studio® 6.0, you cannot access database diagrams, stored procedures, table designs, or view designs in SQL Server 2000. Visual Studio 6.0 Service Pack 4 allows you to alter database diagrams, stored procedures, table designs, or view designs, but you cannot save them. A future Visual Studio Service Pack will allow a limited ability to save changes.

The SQL Server 2000 tools cannot access database diagrams saved using the design tools in Visual Studio 6.0 until you have modified the **dtproperties** table in the database. For more information, see [Backward Compatibility](#).

## See Also

[Editions of SQL Server 2000](#)

[SQL Server 2000: Editions and Components](#)

## Operating Systems Supported by the Editions of SQL Server 2000

## Installing SQL Server

## SQL Server 2000: Editions and Components

SQL Server 2000 editions include the Enterprise Edition, the Standard Edition, the Personal Edition, the Developer Edition, and the Evaluation Edition.

**Note** The Microsoft SQL Server 2000 Personal Edition replaces the Microsoft SQL Server version 7.0 Desktop Edition. To install client tools only, you can use the Personal Edition, which is also available when you purchase the Standard and Enterprise Editions of SQL Server 2000.

When you select **SQL Server 2000 Components** on the opening screen, three options appear on the **Install Components** screen:

### **Install Database Server**

Starts SQL Server Setup, with screens for selecting installation options.

### **Install Analysis Services**

Installs Analysis Services on computers processing OLAP cubes. For more information, see [Installing Analysis Services](#).

### **Install English Query**

Installs English Query on computers running English Query applications. For more information, see [Installing English Query](#).

## **Choosing Components and Options to Install**

You may have a database server, an Internet server, or require a database on a client computer. If running database client/server applications you may or may not require a database on your computer. You may need tools to administer a database server, or you may want to run applications that access an instance of SQL Server. Installation choices for these and other SQL Server configurations are described in the following paragraphs.

### **Installing SQL Server on a Database Server**

If installing a database server, install either SQL Server 2000 Enterprise Edition or SQL Server 2000 Standard Edition. If installing a personal database on your

workstation, install SQL Server 2000 Personal Edition. These installations typically include the database engine, the client database management tools, and the client connectivity components.

On a database server, you can install a default instance of SQL Server 2000 relational database engine. You can also install one or more named instances of the SQL Server 2000 database engine. Other than specifying an instance name, the setup choices are similar to those for installing a default instance.

When installing an instance of SQL Server 2000, you must specify whether you want the instance to use failover clustering. For more information, see [Before Installing Failover Clustering](#).

## **Using SQL Server with Client/Server Applications**

For a computer running database client/server applications, such as Microsoft Visual Basic® applications that connect directly to an instance of SQL Server, you have several options:

- If you require a personal database on your client computer, install the Personal Edition of SQL Server. This setup typically installs the client tools and client connectivity components along with the database engine.
- If you do not require a database on your computer, but need to administer an instance of SQL Server on a database server, or plan to develop SQL Server applications, install the option for Client Tools Only. This option includes the client connectivity components. For more information, see [How to install client tools only \(Setup\)](#).
- If you want to only run applications that access instances of SQL Server on database servers, install the connectivity only components. For more information, see [How to install connectivity only \(Setup\)](#).

## **Using SQL Server with an Internet Server**

On an Internet server, such as a server running Microsoft Internet Information Services (IIS), you typically install the SQL Server 2000 client tools. Client tools

include the client connectivity components used by an application connecting to an instance of SQL Server. In addition, the client tools include the utility for configuring the virtual roots needed for applications to access SQL Server through URLs.

After installing the SQL Server client tools, you configure the virtual roots that support accessing an instance of SQL Server through a URL. For more information about configuring the virtual roots, see [Using IIS Virtual Directory Management for SQL Server Utility](#).

**Note** Although you can install an instance of SQL Server on a computer running IIS, this is typically done only for small Web sites that have a single server computer. Most Web sites have their middle-tier IIS system on one server or cluster of servers, and their databases on a separate server or federation of servers. For more information about federations, see [Federated SQL Server 2000 Servers](#).

If some of the Web pages on an Internet server use English Query, you would also install that component.

## Other SQL Server Components

- For distributing SQL Server 2000 with applications, use the SQL Server 2000 Desktop Engine, a stand-alone database engine that independent software vendors can package with their applications. For more information, see [Distributing SQL Server with Applications](#).

**Note** The Desktop Engine has no graphical user interface and is not related to the SQL Server 7.0 Desktop Edition.

In addition to the major components and editions shown on the **Install SQL Server 2000 Components** screen, other editions of SQL Server 2000 are available: SQL Server 2000 Developer Edition, SQL Server 2000 Windows CE Edition, and the SQL Server 2000 Enterprise Evaluation Edition. For more information, see [Features Supported by the Editions of SQL Server 2000](#).

## See Also

[Editions of SQL Server 2000](#)

## Operating Systems Supported by the Editions of SQL Server 2000

## Installing SQL Server

## Installing English Query

English Query is a development tool that works with Microsoft® SQL Server™ 2000. Using English Query, you can create applications that allow users to query a SQL Server database or an Analysis Services database in English. For example, users can ask, "How many widgets were sold in Washington last year?" instead of using the SQL statements:

```
SELECT sum(Orders.Quantity) FROM Orders, Parts
WHERE Orders.State='WA'
      AND Datepart(Orders.Purchase_Date,'Year')='1999'
      AND Parts.PartName='widget'
      AND Orders.Part_ID=Parts.Part_ID
```

When you install English Query, **English Query** is added to the **Microsoft SQL Server** program group on the **Start** menu. **English Query** contains these shortcuts:

- English Query Books Online
- English Query Tutorials
- Microsoft English Query

If English Query is not installed with Microsoft SQL Server 2000, the Help system will access English Query Books Online, Eqdoc.chm, instead of SQL Server Books Online, SQL80.col. However, both documentation files contain essentially the same material about English Query and both provide context-sensitive (F1) Help for English Query. Regardless of the installation scenario, English Query Books Online is available from the English Query program group.

Installation Requirements for English Query are:

- Microsoft Windows® 95, Microsoft Windows 98, or Microsoft Windows NT® version 4.0 or later

- 40 MB of free disk space
- Microsoft Internet Explorer 5.0 or later

**To install English Query**

# Installing SQL Server

# **Installing Analysis Services**

Microsoft® SQL Server™ 2000 Analysis Services includes a powerful server for the construction and analysis of multidimensional data.

**To install Analysis Services**

## Installing SQL Server

## Setting up Windows Services Accounts

On the Microsoft® Windows NT® and Microsoft Windows® 2000 operating systems, Microsoft SQL Server™ and SQL Server Agent are started and run as Windows services. These services appear in the list of installed services in the Services dialog box, available using Windows Control Panel. The table shows each service name and the term used to refer to the default and named instances of SQL Server, as displayed in the Services dialog box.

Service	Name	Term for default instance	Term for named instance
Microsoft SQL Server	SQL Server	MSSQLSERVER	MSSQ\$InstanceName
Microsoft SQL Server Agent	SQL Server Agent	SQLSERVERAGENT	SQLAgent\$InstanceName

For Microsoft SQL Server™ and SQL Server Agent to run as services in Windows, they must be assigned a Windows user account. Typically, both SQL Server and SQL Server Agent are assigned the same user account, either the local system or domain user account. However, you can customize the settings for each service during the installation process. For more information about how to customize account information for each service, see [Services Accounts](#).

**Note** Microsoft Windows 98 does not support Windows services; instead, SQL Server simulates the SQL Server and SQL Server Agent services. It is not required that you create user accounts for these simulated services.

### Using the Local System Account

The local system account does not require a password, does not have network access rights in Windows NT 4.0 and Windows 2000, and restricts your SQL Server installation from interacting with other servers.

### Using a Domain User Account

A domain user account uses Windows Authentication, that is, the same user name and password used to connect to the operating system is also used to connect to SQL Server. A domain user account is typically used because many server-to-server activities can be performed only with a domain user account, for example:

- Remote procedure calls.
- Replication.
- Backing up to network drives.
- Heterogeneous joins that involve remote data sources.
- SQL Server Agent mail features and SQL Mail. This restriction applies if using Microsoft Exchange. Most other mail systems also require clients (the SQL Server and SQL Server Agent services) to be run on accounts with network access.

**Note** Several servers running SQL Server can share the same user account. When setting up replication, it is recommended that a Publisher and all its Subscribers share the same service account for the SQL Server service.

## **Requirements for Domain User Account**

All domain user accounts must have permission to:

- Access and change the SQL Server directory (\Program Files\Microsoft SQL Server\Mssql).
- Access and change the .mdf, .ndf, and .ldf database files.
- Log on as a service.

- Read and write registry keys at and under:
  - HKEY\_LOCAL\_MACHINE\Software\Microsoft\MSSQLServer
    - or- for any named instance:  
HKEY\_LOCAL\_MACHINE\Software\Microsoft\Microsoft SQL Server.
  - HKEY\_LOCAL\_MACHINE\System\CurrentControlset\Services
    - or- for any named instance:  
HKEY\_LOCAL\_MACHINE\System\CurrentControlset\Services
    - HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib.

In addition, a domain user account must be able to read and write corresponding registry keys for these services: SQLAgent\$InstanceName, MSSearch, and MSDTC.

This table shows additional permissions required for certain functionality.

Service	Permission	Functionality
SQL Server	Network write privileges	Write to a mail slot using <b>xp_sendmail</b> .
SQL Server	Act as part of operating system and replace a process level token	Run <b>xp_cmdshell</b> for a user other than a SQL Server administrator.
SQL Server Agent	Member of the Administrators local group	Create CmdExec and ActiveScript jobs belonging to someone other than a SQL Server administrator.  Use the autorestart feature.  Use run-when-idle jobs.
SQL Server	Member of local Power Users or local Administrators group	Add and delete SQL Server objects in the Windows 2000 Active Directory.

## **Changing User Accounts**

To change the password or other properties of any SQL Server–related service after installing SQL Server, use SQL Server Enterprise Manager. If your Windows password expires and you change it, be sure to also revise the SQL Server services settings in Windows. For more information, see [Changing Passwords and User Accounts](#).

### **See Also**

[Creating Security Accounts](#)

[Planning Security](#)

[Services Accounts](#)

# Installing SQL Server

## File Paths for SQL Server 2000

In Microsoft® SQL Server™ 2000, the default location for the installed SQL Server files has changed. For the default instance of SQL Server, the default directory for both program and data files is \Program Files\Microsoft SQL Server\Mssql. You can specify a file path other than the default for both program and data files.

Shared Tools are installed by default at \Program Files\Microsoft SQL Server\80\Tools. This folder contains files shared by all instances of SQL Server 2000, both default and named. Tools include SQL Server Books Online, Dev Tools, and other components.

Setup also installs files in the Microsoft Windows® system directory. The system file location cannot be changed.

### SQL Server Program File Location

The SQL Server program files are located in \Program Files\Microsoft SQL Server\Mssql\Binn.

The program file location is the root directory where Setup creates the folders that contain program files and files that typically do not change as you use SQL Server. Although these files are not read-only, the folders do not contain data, log, backup files, or replication data; therefore, the space requirements for these files should not increase as SQL Server is used.

**Note** Program files cannot be installed on a removable disk drive.

### SQL Server Data File Location

The SQL Server data files are located in \Program Files\Microsoft SQL Server\Mssql\Data.

The data file location is the root directory where Setup creates the folders that contain database and log files, as well as directories for the system log, backup, and replication data. Setup creates database and log files for the **master**, **model**, **tempdb**, **msdb**, **pubs**, and **Northwind** databases. The SQL Server data file path should be located on a drive that has space available for these files to grow.

**Note** Data files cannot be installed on a file system using compression.

## Specifying File Paths

In SQL Server 2000, due to multiple instance options, the instance name is used in addition to the user-specified location for program and data files. For tools and other shared files, however, instance names are not required.

## Default Instance File Path for Program and Data Files

For the default instance of SQL Server, the default SQL Server directory name (Mssql) is used as the default instance name, along with the directory you specify.

For example, if you specify that the SQL Server default instance be installed at D:\MySqlDir, the file paths are:

- D:\MySqlDir\Mssql\Binn (for program files)
- D:\MySqlDir\Mssql\Data (for data files)

## Named Instance File Path for Program and Data Files

For any named instances, the given name of the instance is used with the directory specified.

For example, if you specify that the instance named **MyInstanceA** be installed at D:\MySqlDir, the paths are:

- D:\MySqlDir\MSSQL\$MyInstanceA\Binn (for program files)
- D:\MySqlDir\MSSQL\$MyInstanceA\Data (for data files)

## See Also

[File Locations for Multiple Instances of SQL Server](#)

[Locating Directories and Files](#)

## Installing SQL Server

## **Upgrading an Existing Installation of SQL Server**

You can upgrade from earlier versions to Microsoft® SQL Server™ 2000, and also perform upgrade operations once SQL Server 2000 is installed. Upgrades to SQL Server 2000 from SQL Server version 6.5 and from SQL Server version 7.0 are different operations. SQL Server 6.5 databases (and related information) are converted to SQL Server 2000 formats. An installation of SQL Server 7.0 is overwritten by SQL Server 2000, unless a named instance configuration is installed, allowing SQL Server 7.0 to remain intact.

After an initial installation of SQL Server 2000, other upgrade options are available. If using more than one instance of SQL Server 2000, you can upgrade one instance by adding components, and have different component sets for multiple instances.

### **Upgrading from SQL Server 7.0 to SQL Server 2000**

You can overwrite your existing installation of SQL Server 7.0 by installing a default instance of SQL Server 2000. You can also keep your installation of SQL Server 7.0 intact by installing a named instance of SQL Server 2000. Both operations are performed using the following procedure.

#### **To upgrade from SQL Server 7.0 to SQL Server 2000**

## Installing SQL Server

## **Basic Installation Options**

This section describes basic installation options for SQL Server 2000. Upgrading an existing installation, or creating a new installation on either a local or remote computer is considered a basic installation option.

For more information about cluster maintenance, performing an unattended setup, or rebuilding the registry, see [Advanced Installation Options](#).

# Installing SQL Server

## Entering Information in Basic Setup Screens

Microsoft® SQL Server™ 2000 Setup provides basic and advanced options. There are two options for a local installation:

- Create a new or additional installation
- Upgrade, remove, or add components to an existing installation

If you choose the upgrade option, you have many other choices available. For more information, see [Existing Installation Options](#). In addition, you can select advanced options. For more information about your initial setup choices, see [Installation Selection](#).

For a basic, local installation, select the option for creating a new or additional installation. After entering user and product identification (ID) information in subsequent screens, choose the components to include in this installation of SQL Server 2000. You can select to install either connectivity only, client tools only (which includes connectivity components), or the complete server and client tools option. For more information, see [Installation Definition](#).

If you choose to install the SQL Server relational database with both server and client tools, select either a named instance or the default instance of SQL Server 2000. For more information, see [Instance Name](#).

After selecting the default instance, or choosing to create a named instance, the standard setup type selection screen is presented. For more information, see [Setup Type: Typical, Minimum, or Custom](#).

Other options may be presented while running Setup, depending on the specifics of your system and installation.

### See Also

[Upgrading an Existing Installation of SQL Server](#)

[Multiple Instances of SQL Server](#)

## Installing SQL Server

## Computer Name

The **Computer Name** dialog box in Setup allows you to install Microsoft SQL Server 2000 on your local computer, on a remote computer, or on a virtual server.

All options for installing and upgrading are available on the local computer. Advanced options, including registry rebuild, unattended installation, and upgrading to a cluster are not available on a remote installation. If you are running Setup on a clustered computer, the Virtual Server option is available.

## Options

### Local Computer

By default, the name in the edit box is the local machine name, that is, the computer on which Setup is running. For a local installation, accept the default and click **Next**.

**Note** If you are installing tools only, **Local Computer** will be the only option available on this dialog box.

### Remote Computer

Enter a computer name for a remote installation, or click **Browse** to locate the remote computer.

### Virtual Server

Enter the name of a new or existing Virtual SQL Server to manage.

This option is available only when Microsoft Cluster Service (MSCS) is detected on an Windows NT or Windows 2000 Enterprise operating system.

### Browse

Click the **Browse** button to locate a remote computer.

This button is available only when the **Remote Computer** option is selected.

## See Also

[Installing a Remote Configuration](#)

[Installing a Virtual Server Configuration](#)

[Before Installing Failover Clustering](#)

[Creating a Failover Cluster](#)

## Installing SQL Server

## Installation Selection

The **Installation Selection** screen is an initial screen in Microsoft® SQL Server™ Setup, where you select among three options for running the installation program.

### Options

#### **Create a new instance of SQL Server, or install Client Tools**

Creates a new installation of SQL Server 2000; either a default or named instance. In addition, this option allows you to install only client tools using the compact disc for any edition of SQL Server 2000, on any operating system other than Microsoft Windows® 95.

#### **Upgrade, remove, or add components to an existing instance of SQL Server**

Allows you to upgrade, remove, or add components to an existing instance of SQL Server. Existing instances include installations of earlier versions (SQL Server version 6.5 and SQL Server version 7.0) as well as instances of SQL Server 2000. For more information, see [Existing Installation Options](#).

#### **Advanced Options**

Select advanced options for cluster maintenance, unattended setup, and registry rebuild.

### See Also

[Multiple Instances of SQL Server](#)

[Upgrading an Existing Installation of SQL Server](#)

[Advanced Installation Options](#)

## Installing SQL Server

## Existing Installation Options

The **Existing Installation Options** Setup screen includes choices for working with upgrades from previous versions of Microsoft® SQL Server™, as well as upgrades to SQL Server 2000 components. Options that do not apply to your specific setup do not appear on the screen.

### Options

#### Add components to your existing installation

Allows you to add components to an existing installation of SQL Server 2000.

#### Uninstall your existing installation

Removes an installation (default or named instance) of SQL Server 2000 from your computer. The instance to remove is specified in the **Instance Name** screen.

#### Upgrade your existing installation

This option is available for use with existing installations of SQL Server 7.0 and SQL Server 2000. Depending on the version, edition, and component makeup of your existing installation, selecting this option starts the process for one of the following upgrades:

- Upgrade from SQL Server 7.0 to SQL Server 2000. (If you cannot upgrade client tools, see Upgrade Issues below.)
- Add components to an existing installation of SQL Server 2000. For example, you may have purchased a SQL Server version with more features, or need to install certain components.

**Note** Upgrades from SQL Server 6.5 to SQL Server 2000 are run using the SQL Server Upgrade Wizard, available on the SQL Server **Start** menu.

#### Upgrade your existing installation to a clustered installation

This option is a step in the process of upgrading from a clustered installation of SQL Server 6.5 or SQL Server 7.0 to a clustered installation of SQL Server 2000. First, the earlier version of SQL Server is upgraded to SQL Server 2000. Next, the existing SQL Server 2000 installation can be upgraded to a cluster. For more information, see [Upgrading to a SQL Server 2000 Failover Cluster](#).

## Upgrade Issues

- On a computer running SQL Server 7.0 client tools only, you may encounter the following message when you choose to upgrade your existing installation: "The default instance detected is not able to be upgraded. Please select New Install to upgrade your tools." This issue can occur if you have installed the SQL Server 7.0 client tools by choosing **Custom** in the **Setup Type** dialog box, and then by selecting tools in the components dialog box. In this situation, the existing client tools installation of SQL Server 7.0 cannot be upgraded due to registry issues. Instead, you must re-install SQL Server, by selecting **Create a new instance of SQL Server, or install Client Tools**.
- You can upgrade a beta version of SQL Server 2000 to the final version of the product by using the option to upgrade your existing installation. If you are performing such an upgrade on a computer or a cluster containing multiple instances, you must first close all instances of SQL Server before upgrading.

## See Also

[Upgrading from SQL Server 7.0 to SQL Server 2000](#)

[Upgrading an Existing Installation of SQL Server](#)

[Select Components](#)

## Installing SQL Server

## Installation Definition

Use the **Installation Definition** screen to select the components to include in this installation of SQL Server 2000. If you select Client Tools Only or Connectivity Only, Setup proceeds and no additional choices are required, unless you select components when installing client tools. If you choose to install Server and Client Tools, additional setup screens will appear.

### Options

#### Client Tools Only

Installs only the client relational database management tools. Included in this option are the client tools for administering SQL Server and the client connectivity components. In addition, this option allows you to select other components to install. For more information, see [How to install client tools only \(Setup\)](#).

#### Server and Client Tools

Installs both server and client tools to create a relational database server with administrative capabilities. Selecting Server and Client Tools presents the full range of additional setup options.

For more information about performing a typical installation of a default instance of the database engine, including all client and connectivity components, see [How to install SQL Server 2000 \(Setup\)](#).

**Note** This option is not available if you are installing client tools using a compact disc for an edition of SQL Server that is not supported by your computer's operating system.

#### Connectivity Only

Installs only the relational database client connectivity components, including MDAC 2.6 (Microsoft Data Access Components), a requirement for connecting to SQL Server 2000 named instances. This option provides connectivity tools only, with no choice of client tools or other components. For more information, see [How to install connectivity only \(Setup\)](#).

## **See Also**

[Management Tools](#)

[Server Components](#)

[Client Connectivity](#)

## Installing SQL Server

## **User Information**

The **User Information** Setup screen prompts you to supply your name and company name. These fields are required.

When installing on a network, be sure to supply the name of a user responsible for using or administering the server.

### **See Also**

[Setting Up Windows Services Accounts](#)

## Installing SQL Server

## Instance Name

Use this screen to add and maintain instances of Microsoft® SQL Server™ 2000.

### Options

#### Default

- When selected, a default instance of SQL Server 2000 is installed. Click **Next** to proceed with the install process.
- When cleared, you can install or maintain a named instance of SQL Server 2000.

**Note** If this check box is not enabled, Setup has detected a default instance of SQL Server on this computer. The default instance could be an installation of SQL Server 6.5, SQL Server version 7.0, or it could be the default instance of SQL Server 2000, already installed. Only one installation of SQL Server, any version, can be the default instance at any one time. For more information, see [Multiple Instances of SQL Server](#).

#### Instance Name

Enter a new instance name, or the name of the instance to maintain. Review and follow the rules for instance names.

**IMPORTANT** It is recommended that instance names be kept to less than 10 characters. Instance names can appear in the user interface of various SQL Server and system tools; shorter names are more readable.

#### Instance Naming Rules

- An instance name is not case-sensitive.
- An instance name cannot be the terms *Default* or *MSSQLServer*.

- Instance names must follow the rules for SQL Server identifiers and cannot be reserved keywords.
- Instance names are limited to 16 characters.
- The first character in the instance name must be a letter, an ampersand (&), an underscore (\_), or a number sign (#). Acceptable letters are those defined by the Unicode Standard 2.0, which includes Latin characters a-z and A-Z, in addition to letter characters from other languages.
- Subsequent characters can be:
  - Letters as defined in the Unicode Standard 2.0.
  - Decimal numbers from either Basic Latin or other national scripts.
  - The dollar sign (\$), a number sign (#), or an underscore (\_).
- Embedded spaces or special characters are not allowed in instance names. Neither is the backslash (\), a comma (,), a colon (:), or the at sign (@).

**WARNING** Only characters that are valid in the current Microsoft Windows® code page can be used in instance names in SQL Server 2000. If a Unicode character not supported under the current code page is used, an error occurs.

## **See Also**

[Working with Instances and Versions of SQL Server](#)

[Working with Named and Multiple Instances of SQL Server 2000](#)

[Naming Conventions for Instances of SQL Server 2000](#)

[Reserved Keywords](#)

# Installing SQL Server

## Setup Type: Typical, Minimum, or Custom

When you install the Microsoft® SQL Server™ 2000 Enterprise Edition, SQL Server 2000 Standard Edition, or SQL Server 2000 Personal Edition, SQL Server Setup offers three installation types in the **Setup Type** dialog box. In addition, you can modify the installation location for both program and data files in this dialog box.

### Options

#### Typical

Installs all of SQL Server using the default installation options. This installation is recommended for most users.

#### Minimum

Installs the minimum configuration necessary to run SQL Server. This installation is recommended for users who have computers with minimum available disk space.

#### Custom

Installs SQL Server and allows you to change any or all of the default options. Use a custom installation to select components and subcomponents, or to change settings for collations, services accounts, authentication, or network libraries.

### Destination Folders

The default installation location is C:\Program Files\Microsoft SQL Server\, for both program and data files.

#### Program files

Click **Browse** to select another installation location for the SQL Server program files.

#### Data files

Click **Browse** to select another installation location for the SQL Server data

files.

**CAUTION** It is recommended that program files not be installed on a cluster disk, so that future upgrades to a cluster are possible. If you select a folder on a cluster disk as a destination for SQL Server program files, a message appears requesting another installation path for program files.

When upgrading an installation of SQL Server 7.0 that has previously had program files installed on a cluster disk, a similar message appears: "Setup will move the program files from the cluster disk. Provide a new location for the program files. The drive letter you select must exist on all nodes of the cluster as a local drive so that you can later upgrade to a clustered installation."

For more information, see [Upgrading to a SQL Server 2000 Failover Cluster](#).

## Components for Each Installation Type

When you install SQL Server on Microsoft Windows NT® or Windows 2000, these options are offered for all installation types:

- Windows Services accounts (logon accounts) for SQL Server and SQL Server Agent.
- Whether to start SQL Server and SQL Server Agent automatically each time the computer is restarted.
- Use of various network libraries, or protocols, including TCP/IP Sockets, Named Pipes, and Multiprotocol.

This table lists the types of installations and components that each installation provides.

Component	Typical	Minimum	Custom
Database Server	Yes	Yes	Optional
Upgrade Tools <sup>1</sup>	Yes	No	Optional
Replication Support	Yes	Yes	Optional
Full-Text Search	Yes	Yes	Optional

Client Management Tools	All	None	Optional
Client Connectivity	Yes	Yes	Not an option
Books Online	Yes	No	Optional
Development Tools	Debugger only	None	Choice of tools
Code Samples	None	None	Choice of samples
Collation Settings	Yes	Yes	Choice of settings

1 Upgrade Tools are installed by default only for the default instance of SQL Server 2000, not for any named instances.

## See Also

[Net-Libraries and Network Protocols](#)

## Installing SQL Server

# Select Components

On the **Select Components** screen, you can choose components and subcomponents to install, or to reinstall if not set up initially. Options for a typical installation are selected by default. Select the components to install or reinstall and clear all others.

**Note** You cannot remove components by clearing check boxes on this screen. The only way to remove installed components is to remove SQL Server entirely.

## Options

### Components

Lists the main components of SQL Server

### Sub-components

Lists the sub-components available for the selected component

SQL Server components and respective subcomponents include:

- [Server Components](#)
  - SQL Server
  - Upgrade Tools
  - Replication Support
  - Full-Text Search
  - Debug Symbols
  - Performance Counters
- [Management Tools](#)

- Enterprise Manager
- Profiler
- Query Analyzer
- DTC Client Support
- Conflict Viewer
- [Client Connectivity](#)
- [Books Online](#)
  - Books Online on Disk
- [Development Tools](#)
  - Headers and Libraries
  - MDAC SDKs
  - Backup/Restore API
  - Debugger Interface
- [Code Samples](#)
  - Choice of many code samples

## **See Also**

[How to add components to an instance of SQL Server 2000 \(Setup\)](#)

# Installing SQL Server

# Server Components

These components can be installed from the Server Components category in the **Select Components** dialog box, when running Setup. Server Components are included when the option for Server and Client Tools is selected as an initial installation choice.

## SQL Server

Installs the SQL Server relational database engine and other core tools. If any SQL Server program files are installed, the SQL Server component must be installed.

**Note** When installing the SQL Server component, the Setup program also installs the **bcp**, **isql**, and **osql** utilities, ODBC, OLE DB, and DB-Library.

## Upgrade Tools

Installs the SQL Server Upgrade Wizard, used to upgrade SQL Server 6.5 databases to the current version.

## Replication Support

Installs the scripts and binary files used for replication.

## Full-Text Search

Installs the Microsoft full-text search engine (Microsoft Search service), which extends the ability to search on character columns beyond the basic equality and LIKE operators.

## Debug Symbols

Installs the debug symbols for installations.

## Performance Counters

Installs performance counters for use with installations.

## See Also

[Installation Definition](#)

# Installing SQL Server

# Management Tools

These components can be installed from the Management Tools category in the **Select Components** dialog box, when running Setup. Management tools are included when the option for Client Tools Only is selected as an initial installation choice.

## Enterprise Manager

Used to perform server and enterprise administrative tasks.

## Profiler

Used to monitor, record, and support auditing of Microsoft SQL Server database activity.

## Query Analyzer

Used to enter Transact-SQL statements and procedures interactively. Also provides graphical query analysis in the form of graphical showplans.

## DTC Client Support

Used to extend database transactions across multiple servers.

Microsoft Distributed Transaction Coordinator (MS DTC) coordinates transactions across a network of systems running Microsoft Windows NT®, Microsoft Windows® 98, and Microsoft Windows 95.

## Conflict Viewer

Used to view and, if necessary, change the way synchronization conflicts are resolved.

## See Also

[Installation Definition](#)

[How to install client tools only \(Setup\)](#)

# Installing SQL Server

## Client Connectivity

The client connectivity component is an option in the **Select Components** dialog box in Setup. The client connectivity component is used to communicate between clients and servers, and includes the Microsoft Data Access Components (MDAC) and network libraries for DB-Library, ODBC, and OLE DB.

This component has no subcomponents. Client Connectivity is installed when the option for Connectivity Only is selected as an initial installation choice.

**Note** To connect to a named instance of SQL Server 2000, MDAC 2.6 must be installed on the client computer.

### See Also

[Installation Definition](#)

[Distributing SQL Server with Applications](#)

# Installing SQL Server

## Books Online

This component can be installed from the Books Online category in the **Select Components** dialog box in Setup. The Books Online component includes both the full SQL Server Books Online for SQL Server 2000 and online Help, available by clicking the Help button or pressing the F1 key in dialog boxes and interface elements.

### Books Online on Disk

Installs the complete documentation set on your local drive in the default shared tools locations: \Program Files\Microsoft SQL Server\80\Tools\Books.

**Note** You may want to view information in SQL Server Books Online for Microsoft® SQL Server™ 7.0. For more information, see [How to access SQL Server Books Online for SQL Server 7.0](#).

# Installing SQL Server

## **Development Tools**

These components can be installed from the Development Tools category in the **Select Components** dialog box in Setup. To install development tools, choose a custom installation in the **Setup Type** screen. The Debugger Interface is an exception; it is included when you choose to install a typical installation.

### **Headers and Libraries**

Installs the include (\*.h) files and library (\*.lib) files needed by a C developer to create programs that use OLE DB, ODBC, DB-Library, Open Data Services, SQL-DMO, Embedded SQL for C, and MS DTC. These files are installed in the \Program Files\Microsoft SQL Server\80\Tools\DevTools\Include and the \...\DevTools\Lib directories by default (shared tools location).

### **MDAC SDKs**

Installs MDAC and XML Software Development Kits.

### **Backup/Restore API**

Installs the header files, sample programs, and documentation required by software vendors to develop custom applications to back up and restore Microsoft SQL Server databases.

### **Debugger Interface**

Installs an interface for stored procedure debugging.

# Installing SQL Server

## Code Samples

The samples component is available from the Code Samples category in the **Select Components** dialog box in Setup. This component installs programming sample files used for reference when you write programs for Microsoft® SQL Server™ 2000. These files are installed in folders in the \Program Files\Microsoft SQL Server\80\Tools\DevTools\Samples directory by default (shared tools location). You can install any or all of these samples.

**Note** Sample code for using a virtual device to backup or restore data is included in the Backup/Restore API, a sub-component of the Development Tools component. For more information, see [Development Tools](#).

Option	Name
ADO	Microsoft ActiveX® Data Objects
DBLIB	DB-Library
Desktop	Desktop Engine
DTS	Data Transformation Services
ESQLC	Embedded SQL for C
Misc	Miscellaneous Samples
MSDTC	Microsoft Distributed Transaction Coordinator
ODBC	Open Database Connectivity
ODS	Open Data Services
OLE Automation	OLE Automation
Replication	Replication
Silver	Sample Database Schemas
SQL-DMO	SQL Distributed Management Objects
SQL-NS	SQL Namespace
Utils	Sample Utilities
XML	XML Samples

# Installing SQL Server

## Network Libraries

On the **Network Libraries** screen, you can select network libraries to install for Microsoft® SQL Server™ 2000. Network libraries are used to pass network packets between clients and a server running SQL Server. The network libraries, implemented as dynamic-link libraries (DLLs), perform the network operations required to communicate using specific interprocess communication (IPC) mechanisms.

A server can listen on, or monitor, multiple network libraries at one time. During installation, SQL Server Setup installs all of the Net-Libraries onto the computer and allows you to configure some or all of the Net-Libraries. If a particular Net-Library is not configured, the server cannot listen on that Net-Library. After installation, you can change these configurations using the Server Network utility.

For a clustered installation, only Named Pipes and TCP/IP are available. When installing a clustered instance, the unsupported network libraries are unavailable. When you install named instances, the Multiprotocol, AppleTalk, and Banyan VINES protocols are unavailable.

## Options

### Named Pipes

Named Pipes support is required on Microsoft Windows NT® and Microsoft Windows® 2000 installations of SQL Server. Server-side Named Pipes is not supported on Microsoft Windows 98. By default, SQL Server listens on the standard pipe for Named Pipes Net-Library connections.

### Named Pipes name

Paths for the default and named instances differ:

- Default instance: \\.\pipe\sql\query
- Named instance: \\.\pipe\MSSQL\$instancename\sql\query

After SQL Server is installed, you can change the pipe name.

## **TCP/IP Sockets**

This Net-Library allows SQL Server to communicate by using standard Windows Sockets as the IPC method across the TCP/IP protocol. By default, all installations of Microsoft SQL Server 2000 on all operating systems use the TCP/IP Sockets Net-Library.

Note the following when using TCP/IP Sockets:

- SQL Server uses UDP port 1434 to establish connections from SQL Server 2000 clients. This socket number is also reserved for SQL Server by Internet Assigned Number Authority (IANA).
- Do not use dynamic ports and do not set a proxy server address, because the port you are listening on can change at each service startup.

## **Port Number**

If you set SQL Server to listen on TCP/IP, type the TCP/IP port number in the Port number box only if you want SQL Server to listen on a port address different from the default address. This is the port that SQL Server listens on when accepting connections from TCP/IP Sockets clients. The default number for a default instance is 1433, the official IANA socket number for SQL Server. The port for a named instance is dynamically assigned when the instance is first started, unless you set an alternate port during setup.

## **Remote Winsock proxy address**

If you set SQL Server to listen on a proxy server using Microsoft Proxy Server over TCP/IP Sockets, type the proxy server address in the Remote WinSock proxy address box when you set up the TCP/IP Sockets Net-Library.

## **Multiprotocol**

The Multiprotocol Net-Library uses the Windows NT remote procedure call (RPC) facility. In addition, the Multiprotocol Net-Library:

- Communicates over most IPC mechanisms supported by Windows NT. Only TCP/IP Sockets, NWLink IPX/SPX, and Named Pipes are

considered tested and supported.

- Allows the use of Windows Authentication over all protocols that RPC supports.
- Supports encryption for user password authentication as well as data.
- Offers performance comparable to native IPC Net-Libraries for most applications.

### **Enable Multiprotocol encryption**

Use Multiprotocol encryption only for compatibility with existing systems. The Secure Sockets Layer (SSL) encryption that can be enabled using the Server Network Utility (after running Setup) is a more comprehensive encryption solution. Multiprotocol encryption is not supported on Windows 98 servers.

**Note** The Multiprotocol Net-Library is not supported with named instances.

### **NWLink IPX/SPX**

This Net-Library allows SQL Server to communicate using the NWLink IPX/SPX protocol.

### **Novell Bindery Service Name**

If you set up SQL Server to listen on NWLink IPX/SPX, the Setup program prompts you for the Novell Bindery service name in which to register SQL Server on the Novell network. The default service name is the computer name of the server computer. The Net-Library allows Novell SPX clients to connect to SQL Server.

The server NWLink IPX/SPX Net-Library is not available on Windows 98 and Windows 95.

### **AppleTalk ADSP**

The server AppleTalk (ADSP) Net-Library allows Apple Macintosh® clients to connect to SQL Server using native AppleTalk (as opposed to TCP/IP

Sockets).

**Note** The AppleTalk Net-Library has not been enhanced for SQL Server 2000 and runs at a SQL Server 7.0 level of functionality. This Net-Library will not be supported in a future release of SQL Server 2000 and is not supported on named instances.

### **Apple Talk Service Object**

If you set up SQL Server to listen on AppleTalk, Setup prompts you for the AppleTalk service object name. The AppleTalk service object name is assigned by your system administrator. It is not necessary to enter an AppleTalk zone because the local zone is used when registering the service.

The AppleTalk Net-Library is not supported on Windows 98 and Windows 95.

### **Banyan VINES**

SQL Server supports Banyan VINES Sequenced Packet Protocol (SPP) as the IPC method across the Banyan VINES IP network protocol. Banyan VINES support for clients and servers running Windows NT is available for SQL Server on the Intel® platform only; it is not available on Windows 98 and Windows 95.

**Note** The Banyan VINES Net-Library has not been enhanced and runs at a SQL Server 7.0 level of functionality. This Net-Library will not be supported in a future release of SQL Server 2000 and is not supported on named instances.

### **Street Talk Service name**

If you set up SQL Server to listen on Banyan VINES, the Setup program prompts you for a StreetTalk service name. This has the form *servicename@group@org*, where *servicename* is the StreetTalk computer-based service name used by SQL Server, *group* is the group, and *org* is the organization. The computer-based service name used by SQL Server must first be created by using the MSERVICE program included with your Banyan VINES software. Also, to start SQL Server, you must be logged in with administrative permissions.

### **Enable protocol encryption for all libraries**

Select this check box to enable protocol encryption for all network libraries.

To use protocol encryption, you must have a certificate on the server. For information about obtaining a certificate, see the Microsoft Windows documentation. If you do not have a certificate, you can enable encryption after installing SQL Server using the Server Network Utility.

## Default Net-Library Settings

**Note** TCP/IP networking must be enabled before running SQL setup.

All Net-Libraries are installed by the Setup program. The table shows the default server and client Net-Library settings by operating system.

Operating system	Server Net-Library settings	Client Net-Library settings
Windows 98	TCP/IP Sockets, Shared Memory	TCP/IP Sockets
Windows 95	Not applicable	TCP/IP Sockets
Windows NT 4.0 (Server and Workstation)	TCP/IP Sockets, Shared Memory, Named Pipes	TCP/IP Sockets, Named Pipes
Windows 2000 (all versions)	TCP/IP Sockets, Shared Memory, Named Pipes	TCP/IP Sockets, Named Pipes

## See Also

[Configuring Client Net-Libraries](#)

[Net-Libraries and Network Protocols](#)

# Installing SQL Server

## Services Accounts

Use the **Services Accounts** screen in Setup to assign a logon account to each of the two Microsoft® SQL Server™ services, SQL Server and SQL Server Agent. Either the local system or the domain user account is used, and you can use the same account for each service. The default setting is to use the same account for each service, and to automatically start each service. To use the default setting, enter your domain password and click **Next**.

You can also customize settings for each service. You can enter one logon account for both services, or specify an account for each. To later change options set on the **Services Accounts** screen, run the Services application in Windows Control Panel.

**IMPORTANT** To create or maintain a Microsoft SQL Server™ 2000 failover cluster, you must be logged on to the computer with administrator privileges, that is, be a member of the **Administrators** local group of the computer or domain. For clustering this means that you must be an administrator of all nodes of the cluster.

When running SQL Server 2000 on Microsoft Windows NT 4.0, in addition to being logged on as an administrator, you must configure both SQL Server and SQL Server Agent to run as administrator accounts.

## Options

**Use the same account for each service. Auto start SQL Server Service.**

The default option: One account is used for both SQL Server and SQL Server Agent. These services start automatically when the operating system starts.

**Customize the settings for each service.**

Allows you to use different settings for the two services.

## Services

Select a service for which you want to customize settings.

**SQL Server**

Select this option to customize settings for the service, Microsoft SQL Server.

### **SQL Server Agent**

Select this option to customize settings for the service, Microsoft SQL Server Agent.

### **Service Settings**

Select service settings as required.

#### **Use the Local System account**

The local system account does not require a password, does not have network access rights in Windows NT 4.0, and may restrict your SQL Server installation from interacting with other servers.

**Note** In Windows 2000, the local system account does allow network access.

#### **Use a Domain User account**

A domain user account uses Windows Authentication to set up and connect to SQL Server. By default, account information appears for the domain user account currently logged on to the computer.

#### **Username**

Accept or change the domain username.

#### **Password**

Enter the domain password.

#### **Domain**

Accept or change the domain name.

#### **Auto Start Service**

Select this option to automatically start a service when your operating system starts. This option is available only when customizing the settings for each service.

The SQL Server Agent service is dependent on the SQL Server service in that you can autostart the SQL Server Agent service only if you autostart the

SQL Server service as well.

**Note** When you click **Back** in the **Services Accounts** dialog box, the window you return to reverts to the default options. Options specified earlier are not retained.

## **See Also**

[Setting Up Windows Services Accounts](#)

[Changing Passwords and User Accounts](#)

## Installing SQL Server

# Authentication Mode

Use this screen to choose the security (authentication) mode you want to use for this installation of Microsoft® SQL Server™ 2000. If you select **Mixed Mode**, you are prompted to enter and confirm the system administrator password. After successful connection to SQL Server, the security mechanism is the same for both modes.

## Options

### Windows Authentication Mode

When a user connects through a Microsoft Windows® user account, SQL Server validates the account name and password using information in the Windows operating system.

### Mixed Mode (Windows Authentication and SQL Server Authentication)

Allows users to connect using Windows Authentication or SQL Server Authentication. Users who connect through a Microsoft Windows user account can make use of trusted connections (connections validated by Windows) in either Windows Authentication Mode or Mixed Mode. SQL Server Authentication is provided for backward compatibility.

### Add password for the sa login

Enter and confirm the system administrator password.

### Blank Password (not recommended)

If a user attempts to connect to an instance of SQL Server providing a blank login name, SQL Server uses Windows Authentication. Additionally, if a user attempts to connect to an instance of SQL Server configured for Windows Authentication Mode using a specific login, the login is ignored and Windows Authentication is used.

## See Also

[Adding a SQL Server Login](#)

[Assigning an sa Password](#)

[Authentication Modes](#)

[Creating Security Accounts](#)

# Installing SQL Server

## **Choose Licensing Mode**

Use this dialog box to set the licensing mode enabling your clients to access this instance of Microsoft® SQL Server™. SQL Server 2000 supports two client access licensing modes, one for each device and another for each processor.

A device in this context can be a workstation, terminal, or any other device running a SQL Server application connected to an instance of SQL Server.

A processor refers to a central processing unit (CPU) installed on a computer running an instance of SQL Server 2000. One computer may have multiple processors installed, requiring multiple processor licenses.

Once a licensing mode is set, you cannot change modes. You can add device or processor licenses after installing SQL Server, using the SQL Server 2000 Licensing Setup utility in Control Panel. (Do not confuse this licensing utility with the Windows Licensing utility, also found in Control Panel.)

For more information about licensing modes, see the Microsoft license agreement for SQL Server 2000.

## **Options**

### **Licensing Mode**

If accessing this dialog box from Control Panel, the mode chosen during setup is selected by default, along with the number of devices or processors you have previously selected.

### **Per Seat for**

The Per Seat licensing mode requires a Client Access License for each device that will access SQL Server 2000 Server. Per Seat is often more economical for networks in which clients connect to more than one server.

In the edit box, select the number of devices to license.

### **Processor License for**

With Processor licensing, a license is needed for each processor installed on the computer running SQL Server. The Processor License allows any number

of devices to access the server, whether through an Intranet or over the Internet.

Using Processor licensing, SQL Server 2000 can take advantage of each installed processor, and support an unlimited number of client devices. A customer that provides access to SQL Server databases over the Internet, or that has a large number of users, will generally choose the Processor License.

In the edit box, select the number of processors to license.

### **Continue**

Click the **Continue** button to complete the installation process, or after modifying the number of devices or processors you want to license.

# Installing SQL Server

## Installing a Remote Configuration

Microsoft® SQL Server™ 2000 can be installed on a remote computer, that is, a computer other than the one on which Setup is running. Before performing a remote installation:

- Ensure that the local and remote computers are running Microsoft Windows NT® or Windows® 2000.
- Ensure that the local and remote computers have an Intel®-compatible processor.
- Ensure that you are logged on to the local computer with a user account that has administrative privileges on the remote computer.

A remote setup is much like a normal installation, with two additional dialog boxes:

- The **Remote Setup Information** dialog box, which is also used when Setup is run on a computer that is part of a cluster. For more information, see [Remote Setup Information](#).
- The **Select Computer** dialog box, which allows you to select a remote computer from the list of computers within the connected domains. The list may include computers not available for this installation, because permission must be granted before installing on a remote computer.

A computer network name may be entered instead of choosing from the list.

### See Also

[Computer Name](#)

## Installing SQL Server

## Remote Setup Information

Remote setup information is required to define security in two different setup situations:

- When you choose to install Microsoft® SQL Server™ 2000 on a remote computer.
- When Setup is run on a computer that is part of a cluster, even if you are not creating or maintaining a failover cluster installation of SQL Server.

For a remote installation, SQL Server Setup collects the information you enter in Setup dialog boxes, recording the entries into the Setup.iss file. At the same time, the remote setup process starts a remote service, copies files to the \admin\$ share directory, and runs an unattended installation on the remote computer using the options specified in Setup.iss.

For clustered computers, the **Remote Setup Information** box is displayed because any installation on a failover cluster system needs the administrator account to install Microsoft Distributed Transaction Coordinator on both nodes, or to verify the presence of MS DTC. Administrator information must be entered that is valid for all selected nodes in the failover cluster system.

## Options

### Username, Password, and Domain

Specify the user account under which SQL Server Setup starts a service on the remote computer. This user account must be an administrator on the remote computer and have read access to the Setup source files directory.

Do not confuse the user account entered on this screen with:

- The user account logged on to the local computer.
- The user account assigned in SQL Server Setup to the SQL Server and SQL Server Agent services.

Each of these user accounts is specified separately. However, you can use the same user information in each case. That is, you can use the same name, password, and Windows domain for each account.

### **Target computer**

The name of the remote computer entered in the **Computer Name** dialog box is shown in static text.

### **Target path**

The name of the remote computer and, in Universal Naming Convention format, the directory on the remote computer where SQL Server is to be installed. For example:

`\\target_computer\C$\Program Files\Microsoft SQL Server`

### **Setup Source Files**

Location of the setup program files used for the remote installation.

### **To perform a remote installation**

## Installing SQL Server

# Advanced Installation Options

When you select the **Advanced** option in the **Installation Options** Setup screen, the **Advanced Options** dialog box provides three choices.

## Options

### **Record Unattended .ISS file**

Create a setup initialization file for unattended installations.

### **Registry Rebuild**

Rebuild registry for a corrupted installation.

### **Maintain a virtual server for failover clustering**

Make changes to existing clusters, such as revising the name, or adding and removing cluster nodes.

## See Also

[Performing an Unattended Installation](#)

[Rebuilding the Registry](#)

[Installing a Virtual Server Configuration](#)

[SQL Server Language Support](#)

## Installing SQL Server

## **Installing a Virtual Server Configuration**

The topics in this section provide information about the Setup screens used in setting up and maintaining failover clustering.

# Installing SQL Server

# Failover Clustering: Defining the Virtual Server

Use the **Failover Clustering** screen to define the virtual server for a new cluster, or to maintain the virtual server definition for an existing cluster. You can add and remove IP addresses; multiple IP addresses are allowed for each virtual server.

## Options

### Virtual Server Name

Displays the network name of the virtual server. This is the name users will see when they connect to the virtual server.

When upgrading to a cluster, this name is entered in the **Virtual Server Name** dialog box.

### IP address

Enter the IP address or addresses used to connect to the virtual server.

### SubNet

Displays the Subnet, which is supplied by MSCS.

### Network

Displays the Network name you assigned each subnet during setup of MSCS.

### Add

Adds the specified IP address and SubNet to the named virtual server.

### Remove

Removes the specified IP address and SubNet from the named virtual server.

## See Also

[Before Installing Failover Clustering](#)

[Creating a Failover Cluster](#)

## [Upgrading to a SQL Server 2000 Failover Cluster](#)

## Installing SQL Server

# Cluster Management Screen

Use the **Cluster Management** screen to review the cluster definition provided by Microsoft® SQL Server™ 2000, and make changes if necessary. After you have specified nodes for the virtual server, Setup installs or uninstalls the SQL Server binary files on each node.

**CAUTION** If you modify the node list of a virtual server using the quorum resource, your cluster may not fail over properly. For more information, see [Modify Node List Warning](#).

## Options

### Available Nodes

A list of computers that can be added to the current virtual server definition. If a computer you want is not available at this time, you can run Setup later to add it to the virtual server definition.

### Configured Nodes

List of computers currently configured in the current virtual server definition. The computer at the top of the list is the preferred node.

### Unavailable Nodes

Computers that are currently offline or not available to be added to a cluster definition.

### Add

Adds the selected available node to the list of configured nodes.

### Remove

Removes the selected configured node from the list of configured nodes.

## See Also

[Failover Clustering](#)

[Maintaining a Failover Cluster](#)

[Cluster Disk Selection Screen](#)

[Upgrading to a SQL Server 2000 Failover Cluster](#)

# Installing SQL Server

## Cluster Disk Selection Screen

Use the **Cluster Disk Selection** screen to select a cluster group during the installation of a new virtual server or during an upgrade to a cluster. A cluster group is composed of one or more shared cluster disks within a group, and can contain at most one Microsoft® SQL Server™ virtual server. The **Cluster Disk Selection** screen lists only those groups that already have the shared cluster disk added as a resource. For more information about cluster disks, see [Creating a Failover Cluster](#).

**CAUTION** Do not select the quorum disk (the last group in the list) because the quorum disk must be treated as a special resource. Clustering may fail if selected as a cluster group. A warning message appears if you select the quorum disk. For more information, see [Quorum Disk Selection Warning](#).

When using a small cluster, the quorum disk may be the only choice available. Use it only for testing purposes or to explore failover clustering.

**IMPORTANT** Never use the quorum group for production purposes.

### See Also

[Failover Clustering](#)

[Maintaining a Failover Cluster](#)

[Modify Node List Warning](#)

## Installing SQL Server

## Quorum Disk Selection Warning

The following warning message appears if you select the quorum disk (the last group in the list) on the **Cluster Disk Selection** screen. This warning applies to both Microsoft® Windows NT® 4.0 and Microsoft Windows® 2000:

It is strongly recommended that you not use the quorum group with SQL Server.

The quorum disk is a special resource in the Windows operating system. If you select the quorum disk, you may later want to restrict ownership of Microsoft® SQL Server™ to a subset of the cluster nodes. However, the quorum group owner list must include all of the nodes in the cluster.

For example, you may have a two-node cluster (Node1 and Node2) with SQL Server set to use the quorum disk group. If you then modify SQL Server to have only Node1 in the virtual server definition, the quorum disk group is prevented from failing over to Node2. In the event of a failure of Node1, the result is that you not only lose the virtual SQL Server, but the entire MSCS cluster.

This is true for Windows NT 4.0 only. In Windows 2000 the node list is ignored and the quorum disk group can fail over to any node in the cluster configuration. However, another issue may arise. In the previous example, there are no SQL Server program files available on Node2, but the cluster group can fail over to Node2. In this situation, SQL Server is unable to run on Node2 but the cluster group containing the quorum disk may fail over anyway, making your SQL Server unavailable.

For more information about the quorum disk, see the Windows NT documentation.

### See Also

[Cluster Disk Selection Screen](#)

[Modify Node List Warning](#)

[Creating a Failover Cluster](#)

## Failover Clustering Dependencies

# Installing SQL Server

## Modify Node List Warning

In the **Cluster Management** screen, use caution if you modify the node list of a Microsoft® SQL Server™ 2000 virtual server using the quorum resource. If such a node list is modified, the following warning appears when **Next** is clicked:

Modifying the node list of the quorum resource may prevent your cluster from failing over properly. Are you sure you want to do this?

The quorum resource itself is unable to fail over to any servers that you did not select as part of your virtual server definition. This may jeopardize the availability of your failover cluster. For more information, see the Microsoft Windows NT® documentation.

**Note** This problem does not occur when you run SQL Server 2000 on Microsoft Windows® 2000.

### See Also

[Quorum Disk Selection Warning](#)

[Creating a Failover Cluster](#)

[Failover Clustering Dependencies](#)

## Installing SQL Server

## Performing an Unattended Installation

You can perform an unattended installation of Microsoft® SQL Server™ 2000, in which setup screen entries are made automatically using stored information. An unattended installation can be convenient if you want to perform several installations of SQL Server with identical configurations on different computers. An unattended installation requires a setup initialization file, which can be created in several different ways.

By default, each time you install SQL Server using the Setup screens, the options you select are recorded into the setup initialization file, Setup.iss. Setup.iss is placed in the system root directory (%windir%), and is available to provide installation settings at a later time.

**Note** You cannot perform an unattended installation to set up a failover cluster of Microsoft SQL Server 2000.

### Creating a Setup File Using the Record Unattended Option

In Setup, when you select the **Record Unattended .ISS file** option in the **Advanced Options** screen, each subsequent choice you make in the setup screens is recorded in the Setup.iss file stored in the system root directory. SQL Server files are not installed in this process. The Setup.iss file can then be run as is, or revised in a text editor if necessary.

**To record an unattended installation file**

# Installing SQL Server

## Creating a Setup File Manually

You create a customized setup initialization file interactively when you select the **Record Unattended** option in Microsoft® SQL Server™ 2000 Setup. You can also edit files manually, to further refine and customize setup initialization files.

## Creating or Modifying a Setup File Using a Text Editor

You can use a text editor to modify the Setup.iss file generated using the **Record Unattended .ISS file** option. You can also modify one of the sample setup files (\*.iss) included on the SQL Server 2000 compact disc or you can create your own setup file.

To modify one of the sample setup initialization files found on the SQL Server compact disc, open the file in a text editor and modify as required. Keep the file compatible with the Microsoft Windows® initialization file format and save it with the .iss file name extension.

## Format of a Sample Setup Initialization File

A setup initialization file is a text file that uses the standard Windows .ini file format. Sections of the sample setup initialization file for a typical installation of Microsoft SQL Server 2000 are described in the tables that follow. This sample file (Sqlins.iss) is found in the root directory of the SQL Server compact disc.

**Note** In creating a setup file for a named instance of SQL Server 2000, you must indicate the instance name you want to install and the path required to navigate through the setup screens. The **Instance Name** dialog box [DlgInstanceName] section must be modified, as well as other places in the setup file where the instance name appears.

### [InstallShield Silent]

This section is required for InstallShield. Do not change the values.

Entry	Value	Description
Version	v5.00.000	Version of the InstallShield

		Silent response file.
File	Response File	Indicates this is the Response File.

## [File Transfer]

Entry	Value	Description
OverwriteReadOnly	NoToAll	Do not overwrite read-only files.

## [DlgOrder]

Lists each dialog box in the order it appears in an attended setup. The listing in this section must correspond to the other sections in the setup initialization file.

Entry	Value	Description
Dlg0	SdWelcome-0	Initial dialog box
Count	14	Number of dialog boxes listed in this section
Dlg1	DlgMachine-0	Next dialog box
Dlg2	DlgInstallMode-0	Next dialog box
Dlg3	SdRegisterUser-0	Next dialog box
Dlg4	SdLicense-0	Next dialog box
Dlg5	CDKEYDialog-0	Next dialog box
Dlg 6	DlgClientServer-0	Next dialog box
Dlg7	DlgInstanceName-0	Next dialog box
Dlg8	SetupTypeSQL-0	Next dialog box
Dlg9	DlgServices-0	Next dialog box
Dlg10	DLGSqlSecurity-0	Next dialog box
Dlg11	DlgCollation-0	Next dialog box
Dlg12	DlgServerNetwork-0	Next dialog box
Dlg13	SdStartCopy-0	Next dialog box
Dlg14	SdFinish-0	Last dialog box

### [SdWelcome-0]

Corresponds to the **Welcome** dialog box.

Entry	Value	Description
Result	1	Next

### [DlgMachine-0]

Corresponds to the **Computer Name** dialog box.

Entry	Value	Description
Type	1	Local computer
Result	1	Next

### [DlgInstallMode-0]

Corresponds to the **Installation Selection** dialog box.

Entry	Value	Description
Type	1	Create a new instance
Result	1	Next

### [SdRegisterUser-0]

Corresponds to the **User Information** dialog box.

Entry	Value	Description
szName	<user name>	Name of user; company name is not required.
Result	1	Next.

### [SdLicense-0]

Corresponds to the **Software License Agreement** dialog box.

Entry	Value	Description
Result	1	Yes

### [CDKEYDialog-0]

Corresponds to the **CD-Key** dialog box.

Entry	Value	Description
svCDKey	<CD key value>	Specified for each installation
Result	1	Next

### [DlgClientServer-0]

Corresponds to the **Installation Definition** dialog box.

Entry	Value	Description
Type	2	Server and client tools
Result	1	Next

### [DlgInstanceName-0]

Corresponds to the **Instance Name** dialog box.

Entry	Value	Description
InstanceName	MSSQLSERVER	Designation of the default instance (always the same).

InstanceName	<instance name>	Designation of a named instance.
Result	1	Next.

### [SetupTypeSQL-0]

Corresponds to the **Setup type** dialog box.

Entry	Value	Description
szDir	%PROGRAMFILES%\Microsoft SQL Server	Directory where SQL Server program files are installed.
Result	301	Typical (301) (302 = Minimum and 303 = Custom).
szDataDir	%PROGRAMFILES%\Microsoft SQL Server	Directory where SQL Server data files are installed (same as program files).

### [DlgServices-0]

Corresponds to the **Services Accounts** dialog box.

Entry	Value	Description
Local-Domain	3855 <other numeric value>	Use the same account for each service.  To customize the settings for each service, see <a href="#">Setup Initialization File Details</a> .
AutoStart	15	Autostart Service is enabled.
Result	1	Next.

---

### [DlgSQLSecurity-0]

Corresponds to the **Authentication** dialog box. Choices shown here include options not in the sample Sqlins.iss file.

Entry	Value	Description
LoginMode	-1	System default security is used.  The Microsoft Windows NT® default is Windows Authentication Mode.  The Microsoft Windows 98 default is Mixed Mode, with no <b>sa</b> password.
LoginMode	1	Windows Authentication Mode.
LoginMode	2	Mixed Mode.
szPwd	<choice of password>	Used only with Mixed Mode security.
Result	1	Next.

### [DlgCollation-0]

Corresponds to the **Collation Settings** dialog box.

Entry	Value	Description
collation_name	' '	When blank, system default collation is used.
collation_name	<collation designator>	Selected by user. For more information, see <a href="#">Collation Settings in Setup</a> .

### [SdServerNetwork-0]

Corresponds to the **Network Libraries** dialog box.

Entry	Value	Description
NetworkLibs	255 245 15	Named pipes and TCP/IP Value 245 = TCP/IP only Value 15 = Named pipes only To customize network library settings, see <a href="#">Setup Initialization File Details</a>
TCPPort	1433	Port address, for TCP/IP
TCPPrxy	Default	Default proxy, or what is entered
NMPPipeName	\\.\pipe\sql\query	Pipe name
Result	1	Next

### [SdStartCopy-0]

Corresponds to the **Start Copying Files** dialog box.

Entry	Value	Description
Result	1	Next

### [SdFinish-0]

Corresponds to the **Setup Complete** dialog box.

Entry	Value	Description
Result	1	Next
bOpt1	0	Placeholder for stock dialog box
bOpt2	0	Placeholder for stock dialog box

		box
--	--	-----

## **See Also**

[Performing an Unattended Installation](#)

[Setup Initialization File Details](#)

# Installing SQL Server

## Setup Initialization File Details

When creating a customized setup initialization file, the **Service Accounts** and **Network Libraries** dialog boxes have additional options used for an unattended installation.

### Services Accounts Dialog Box

To customize settings for each service, you can calculate values for the Local-Domain and AutoStart entries.

#### Local-Domain

If you want the SQL Server and SQL Server Agent services to use different logon accounts, you can calculate the value to enter for Local-Domain. The Local-Domain value is a bitwise logical OR combination of the values shown in the following table. For more information, see [|\(Bitwise OR\)|](#).

Service	Account to use	Hexadecimal value
SQL Server	<b>Local System</b> account	x0000000F
SQL Server Agent	<b>Local System</b> account	0x00000F00
SQL Server	<b>Domain User</b> account	0x000000F0
SQL Server Agent	<b>Domain User</b> account	0x0000F000

Additional entries must be added to your setup initialization file if the **Domain User** account is used for either service.

When SQL Server service is using a domain account, Setup looks for values for:

- SQLDomain = <domain name>
- SQLDomainAcct = <domain user account>
- SQLDomainPwd = <domain password: an encrypted password available only using setup screens>

When SQL Server Agent service is using a domain account, Setup looks for values for:

- AgtDomain = <domain name>
- AgtDomainAcct = <domain user account>
- AgtDomainPwd = <domain password: an encrypted password available only using setup screens>

## AutoStart

The value for the **AutoStart** option is a bitwise logical OR combination using the following hexadecimal values:

Autostart SQL Server = 0x0000000F

Autostart SQL Server Agent = 0x000000F0

A value of zero (0) for either service indicates no AutoStart; manual startup is required.

## Network Libraries Dialog Box

To customize network library settings, you can calculate a value for the NetworkLibs entry in the **Network Libraries** dialog box. The value for NetworkLibs is a bitwise logical OR combination of the values shown in the following table. When a network library is set, additional information must be entered in the setup initialization file, as shown in the third column.

Network Library to use when connecting to the server	Hexadecimal value to use in Bitwise OR operation	Additional information that Setup looks for
Named Pipes	0xF	NMPPipeName = <named pipe name>
TCP/IP Sockets	0xF0	TCPPort = <port number> TCPPrxy = <Remote

		Winsock proxy address>
NW Link	0xF000	NWLinkObj = <Novell Bindary service name>
Apple Talk	0xF0000	ApplObj = <Apple Talk service object>
Banyan VINES	0xF00000	BanyanObj = <StreetTalk service name>

**See Also**

[Services Accounts](#)

[Network Libraries](#)

[Performing an Unattended Installation](#)

# Installing SQL Server

## Installing SQL Server Using SMS

You can use Microsoft® Systems Management Server (SMS) version 1.2 or later to install Microsoft SQL Server™ 2000 automatically on multiple server computers running Microsoft Windows NT® or Microsoft Windows® 2000 in your enterprise.

The SQL Server compact disc contains a Package Definition Format (PDF) file (Smssql.pdf) that automates creating a SQL Server package in SMS. The SQL Server package can then be distributed and installed on SMS computers.

Smssql.pdf includes instructions for running the batch file Smssqins.bat with Sqlins.iss (the setup initialization file) for a typical installation. Both of these files are included on the SQL Server compact disc.

To create a custom command file, edit a copy of Smssql.pdf.

### See Also

[Performing an Unattended Installation](#)

[Creating a Setup File Manually](#)

# Installing SQL Server

## Rebuilding the Registry

The **Registry Rebuild** option on the **Advanced Options** Setup screen allows you to rebuild the registry for a corrupted Microsoft® SQL Server™ installation. This process fixes only the registry; it does not fix data errors or the **master** database.

**IMPORTANT** To rebuild the registry, you must enter setup information using the same choices that you entered during the initial installation. If you do not know or are not sure of this information, do not use this registry rebuild process. To restore the registry, you must uninstall and reinstall SQL Server.

**To rebuild the registry**

# Installing SQL Server

## Working with Named and Multiple Instances of SQL Server 2000

With Microsoft® SQL Server™ 2000, you have the option of installing multiple copies, or instances of SQL Server on one computer. When setting up a new installation of SQL Server 2000 or maintaining an existing installation, you can specify it as:

- A default instance of SQL Server.

This instance is identified by the network name of the computer on which it is running. Applications using client software from earlier versions of SQL Server can connect to a default instance. SQL Server version 6.5 or SQL Server version 7.0 servers can operate as default instances. However, a computer can have only one version functioning as the default instance at a time.

- A named instance of SQL Server.

This instance is identified by the network name of the computer plus an instance name, in the format `<computername>\<instancename>`.

Applications must use SQL Server 2000 client components to connect to a named instance. A computer can run any number of named instances of SQL Server concurrently. A named instance can run at the same time as an existing installation of SQL Server version 6.5 or SQL Server version 7.0. The instance name cannot exceed 16 characters.

A new instance name must begin with a letter, an ampersand (&), or an underscore (\_), and can contain numbers, letters, or other characters. SQL Server **sysnames** and reserved names should not be used as instance names. For example, the term "default" should not be used as an instance name because it is a reserved name used by Setup.

Single and multiple instances of SQL Server 2000 (default or named) are available using the SQL Server 2000 Personal Edition, the SQL Server 2000 Standard Edition, or the SQL Server 2000 Enterprise Edition.

## Default Instances

You cannot install a default instance of SQL Server 2000 on a computer that is also running SQL Server 7.0. You must either upgrade the SQL Server 7.0 installation to a default instance of SQL Server 2000, or keep the default instance of SQL Server 7.0 and install a named instance of SQL Server 2000.

You can install a default instance of SQL Server 2000 on a computer running SQL Server 6.5, but the SQL Server 6.5 installation and the default instance of SQL Server 2000 cannot be running at the same time. You must switch between the two using the SQL Server 2000 **vswitch** command prompt utility.

## Multiple Instances

Multiple instances occur when you have more than one instance of SQL Server 2000 installed on one computer. Each instance operates independently from any other instance on the same computer, and applications can connect to any of the instances. The number of instances that can run on a single computer depends on resources available. The maximum number of instances supported in SQL Server 2000 is 16.

When you install SQL Server 2000 on a computer with no existing installations of SQL Server, Setup specifies the installation of a default instance. However, you can choose to install SQL Server 2000 as a named instance instead by clearing the **Default** option in the **Instance Name** dialog box.

A named instance of SQL Server 2000 can be installed at any time: before installing the default instance of SQL Server 2000, after installing the default instance of SQL Server 2000, or instead of installing the default instance of SQL Server 2000.

Each named instance is made up of a distinct set of services and can have completely different settings for collations and other options. The directory structure, registry structure, and service names all reflect the specific instance name you specify.

## See Also

[Multiple Instances of SQL Server](#)

[Naming Conventions for Instances of SQL Server 2000](#)

[Network Protocols for Named Instances](#)

[File Locations for Multiple Instances of SQL Server](#)

[Working with Instances and Versions of SQL Server](#)

# Installing SQL Server

# Naming Conventions for Instances of SQL Server 2000

Because Microsoft® SQL Server™ 2000 can be set up to include one or more named instances, with or instead of a default instance, new naming conventions are used to distinguish between instances.

In earlier versions, a SQL Server installation is identified by computer name. In SQL Server 2000, only the default instance is identified solely by computer name. A named instance is identified by a combination of computer name and instance name. This instance name is also reflected in the names of the associated SQL Server services.

**Note** There can be only one default instance of SQL Server for each computer. It can be an intact SQL Server version 6.5 or SQL Server version 7.0 installation, or it can be an installation of SQL Server 2000 set up as the default instance. In either case, the default instance uses the same service names, registry structure, network listening points, and other defaults used in SQL Server 7.0.

## Service Names for Default and Named Instances

When you install a default instance of SQL Server, the service names remain MSSQLServer and SQLServerAgent (the same as in SQL Server 7.0).

When you install a named instance of SQL Server, the service names are changed to:

- MSSQL\$InstanceName for the MSSQLServer service.
- SQLAgent\$InstanceName for the SQLServerAgent service.

The Microsoft Distributed Transaction Coordinator and Microsoft Search services are installed only once, and can be used simultaneously by every installed instance of SQL Server.

## See Also

## Multiple Instances of SQL Server

# Installing SQL Server

## Network Protocols for Named Instances

When you install a default instance of Microsoft® SQL Server™ 2000, the standard network addresses are enabled. For example, named pipes uses `\\.\pipe\sql\query`, and TCP/IP sockets connect to port 1433.

When you select a named instance, only the Named Pipes, TCP/IP, and NWLink IPX/SPX protocols are supported. Named Pipes defaults to a network address of `\\Computername\Pipe\MSSQL$instancename\Sql\Query`. The port addresses used by TCP/IP and NWLink IPX/SPX are chosen dynamically (by default) the first time the instance is started.

### See Also

[Communicating with Multiple Instances](#)

[Multiple Instances of SQL Server](#)

[Network Libraries](#)

# Installing SQL Server

## File Locations for Multiple Instances of SQL Server

Each named instance of Microsoft® SQL Server™ 2000 has a specific location for its program files and another for its data files that is different from that of the default instance of SQL Server.

**Note** A named instance is not necessarily the same as a multiple instance. You can have a single named instance or you can have multiple named instances. For more information, see [Multiple Instances of SQL Server](#).

For each named instance of SQL Server that you install, the default directories are:

- \Program Files\Microsoft SQL Server\MSSQL\$InstanceName\Binn for executable files.
- \Program Files\Microsoft SQL Server\MSSQL\$InstanceName\Data for data files.

Shared tools for all instances, both default and named instances, are located in the \Program Files\Microsoft SQL Server\80\Tools directory. You can specify file paths other than the default locations for program and data file for multiple instances.

The following illustration shows the simplest case of multiple instances of Microsoft SQL Server 2000: the default instance and one named instance, **Instance1**. A named instance has its own full set of data files and executable files. Common files used by both the default instance and any named instances are installed in the folder \Program Files\Microsoft SQL Server\80.

**Note** If Microsoft SQL Server version 7.0 is used as the default installation alongside a named instance of SQL Server 2000, program and data files are located at C:\Mssql7, the default location for SQL Server 7.0 files.



### Finding Install Locations

If you are uncertain about instance paths, query the registry to get the installation

path of a particular instance. Run the following at the command prompt, inserting the appropriate instance name:

```
C:\> REG QUERY HKLM\Software\Microsoft\Microsoft SQL Server\
```

**Note** The REG QUERY tool is available in the Microsoft Windows® 2000 Resource Kit.

## **See Also**

[File Paths for SQL Server 2000](#)

[Multiple Instances of SQL Server](#)

# Installing SQL Server

## **Removing Multiple Instances of SQL Server 2000**

When you remove a default or named instance of Microsoft® SQL Server™ 2000, the data files and registry keys for that instance are deleted. Tools cannot be removed until all instances of SQL Server 2000 have been removed from a computer, because the tools are shared among all installed instances.

To remove a single instance of SQL Server 2000, or to remove all installed instances, see [How to remove SQL Server 2000 \(Windows\)](#).

# Installing SQL Server

## Working with Instances and Versions of SQL Server

Multiple instances in Microsoft® SQL Server™ 2000 offer enhanced ways to work with earlier versions of Microsoft SQL Server already installed on your computer. You can leave previous installations intact, and also install and run SQL Server 2000. For example, you can run SQL Server version 7.0 and a named instance of SQL Server 2000 at the same time, or you can run SQL Server version 6.5 in a version switch configuration with SQL Server 2000. If you need to have three different versions of SQL Server installed on the same computer, there are several ways to accomplish this.

In addition, users of all editions of SQL Server can have more than one instance of SQL Server 2000 installed and running at once (multiple instances), as well as one or more earlier versions.

Considerations for using SQL Server 2000 in combination with previous installations include:

- Using SQL Server 6.5 with the default instance or named instances of SQL Server 2000.
- Running SQL Server 7.0 with a named instance of SQL Server 2000.
- Working with three versions of SQL Server: SQL Server 6.5, SQL Server 7.0, and SQL Server 2000.

**Note** The concept of the default instance is new to SQL Server 2000, due to the introduction of multiple instances. If installed on the same computer as SQL Server 2000, either SQL Server version 6.5 or SQL Server version 7.0 can function as default instances of SQL Server. (A default instance is identified by the network name of the computer on which it is running.) For more information, see [Working with Named and Multiple Instances of SQL Server 2000](#).

### Using SQL Server Books Online for SQL Server 7.0

When you keep Microsoft SQL Server version 7.0 on your computer and install a named instance of SQL Server 2000, SQL Server Books Online for SQL Server 7.0 remains in its original location: C:\Mssql7\Books. In this side-by-side configuration, Books Online for SQL Server 7.0 remains accessible from the start menu in the SQL Server 7.0 program group.

**Note** This is an exception to what occurs for the other shared tools (such as code samples, scripts, and templates), when a named instance of SQL Server 2000 is installed along with SQL Server 7.0. All other shared tools from the 7.0 installation are copied to storage locations, with pointers to the SQL Server 2000 tools replacing previous versions of the tools. Files for Books Online for SQL Server 7.0 are not redirected in this way -- they remain ready for use.

When SQL Server 7.0 is upgraded to the default version of SQL Server 2000, the 7.0 Books Online files are also upgraded. That is, they are replaced with the SQL Server 2000 Books Online.

Whether you have SQL Server 7.0 installed or not, you can access information in the SQL Server 7.0 documentation. For more information, see [How to access SQL Server Books Online for SQL Server 7.0](#).

## **See Also**

[Using SQL Server 6.5 with SQL Server 2000](#)

[Running SQL Server 7.0 Along with a Named Instance of SQL Server 2000](#)

[Working with Three Versions of SQL Server](#)

# Installing SQL Server

## Using SQL Server 6.5 with SQL Server 2000

If you have Microsoft® SQL Server™ version 6.5 installed, you can keep the SQL Server version 6.5 installation and also install a default or named instance of SQL Server 2000. No version upgrading is involved; however, version switching can be used to move between SQL Server version 6.5 and SQL Server 2000. In addition, SQL Server 2000 tools are used to control both SQL Server 2000 and SQL Server version 6.5.

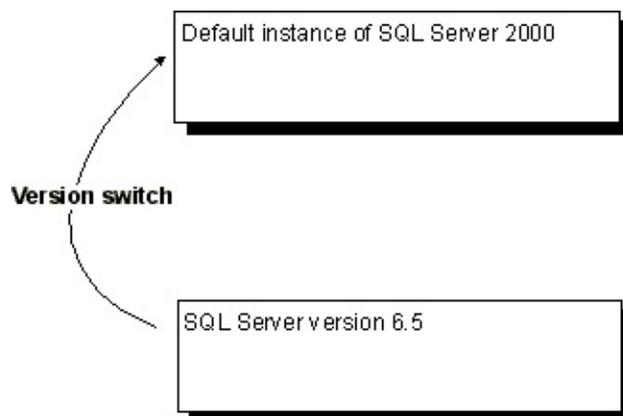
**WARNING** After SQL Server 2000 is installed, the SQL Server version 6.5 Trace utility and other earlier tools are no longer available.

To install SQL Server 2000 alongside SQL Server 6.5:

- Keep your SQL Server 6.5 configuration intact.
- Install SQL Server 2000, selecting either a default or named instance in the **Instance Name** dialog box.
- Switch versions from SQL Server 6.5 to the default instance of SQL Server 2000.

**Note** Switching from SQL Server 2000 back to SQL Server 6.5 is not recommended.

The illustration shows an installation of SQL Server 6.5 in a version switch configuration with SQL Server 2000.



## **See Also**

[Switching Between SQL Server 6.5 and SQL Server 2000](#)

[Running SQL Server 7.0 Along with a Named Instance of SQL Server 2000](#)

[Working with Three Versions of SQL Server](#)

# Installing SQL Server

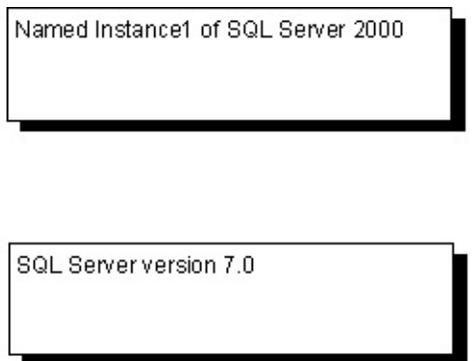
## Running SQL Server 7.0 Along with a Named Instance of SQL Server 2000

You can keep an installation of Microsoft® SQL Server™ version 7.0 intact on your computer and also install a named instance of SQL Server 2000 on the same computer. This configuration enables you to run both the original installation of SQL Server 7.0 and the named instance of SQL Server 2000 at the same time, without using the **vswitch** command prompt utility.

To run a named instance of SQL Server 2000 with an existing SQL Server 7.0 installation intact:

- Keep SQL Server version 7.0 in its original condition with no version upgrade to SQL Server 2000. SQL Server 7.0 functions as the default instance of SQL Server, identified by the network name of the computer.
- Install a named instance of SQL Server 2000, identified by both the network name of the computer plus an instance name.

The illustration shows this configuration.



### See Also

[How to install a named instance of SQL Server 2000 \(Setup\)](#)

[Working with Three Versions of SQL Server](#)

## Installing SQL Server

## **Working with Three Versions of SQL Server**

This topic describes two scenarios for working with SQL Server version 6.5, SQL Server version 7.0, and SQL Server 2000. One example shows three versions installed at one time, with no version upgrades, but with a version switch between SQL Server 6.5 and SQL Server 7.0. The other involves upgrading to SQL Server 2000 from SQL Server 7.0, and then version switching between SQL Server 6.5 and SQL Server 2000.

In any of these situations, multiple named instances of SQL Server 2000 can be installed as well. However, only two different versions of SQL Server can run at one time, using version switching in one of two ways:

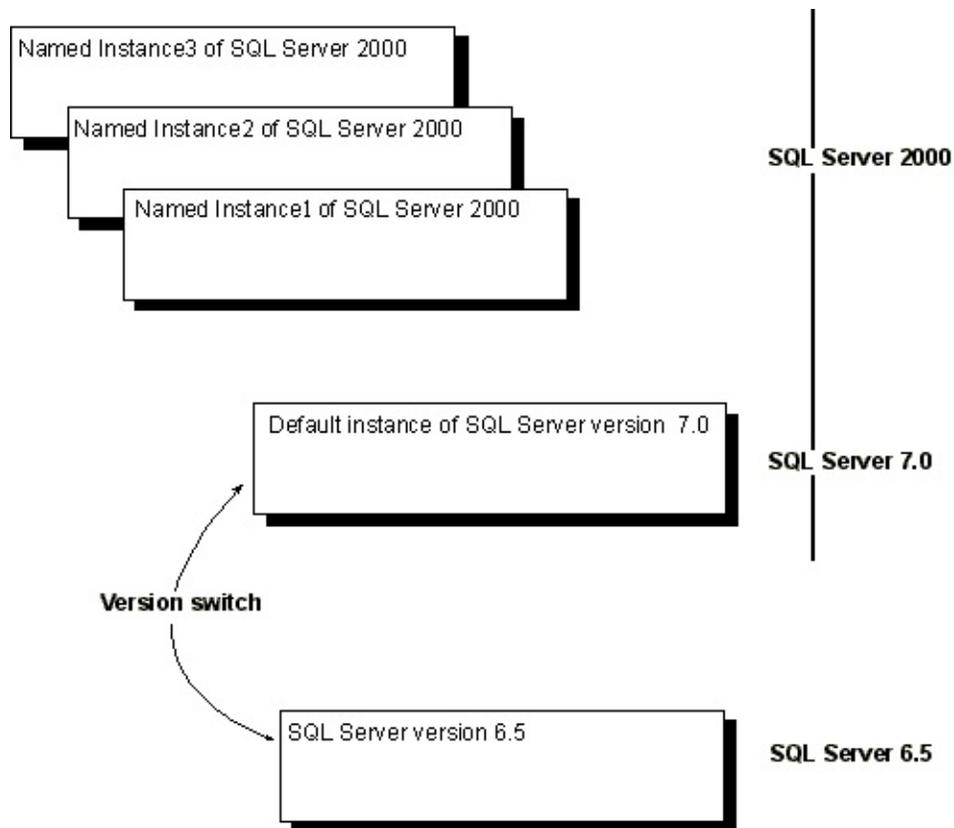
- Switch between SQL Server 6.5 and SQL Server 7.0.
- Switch between SQL Server 6.5 and SQL Server 2000.

### **Using Version Switching**

To use version switching with SQL Server 6.5 and SQL Server 7.0, while at the same time running multiple instances of SQL Server 2000:

- Keep the SQL Server 6.5 configuration intact.
- Keep the SQL Server 7.0 configuration intact, with no version upgrade to SQL Server 2000.
- Install one or more named instances of SQL Server 2000.

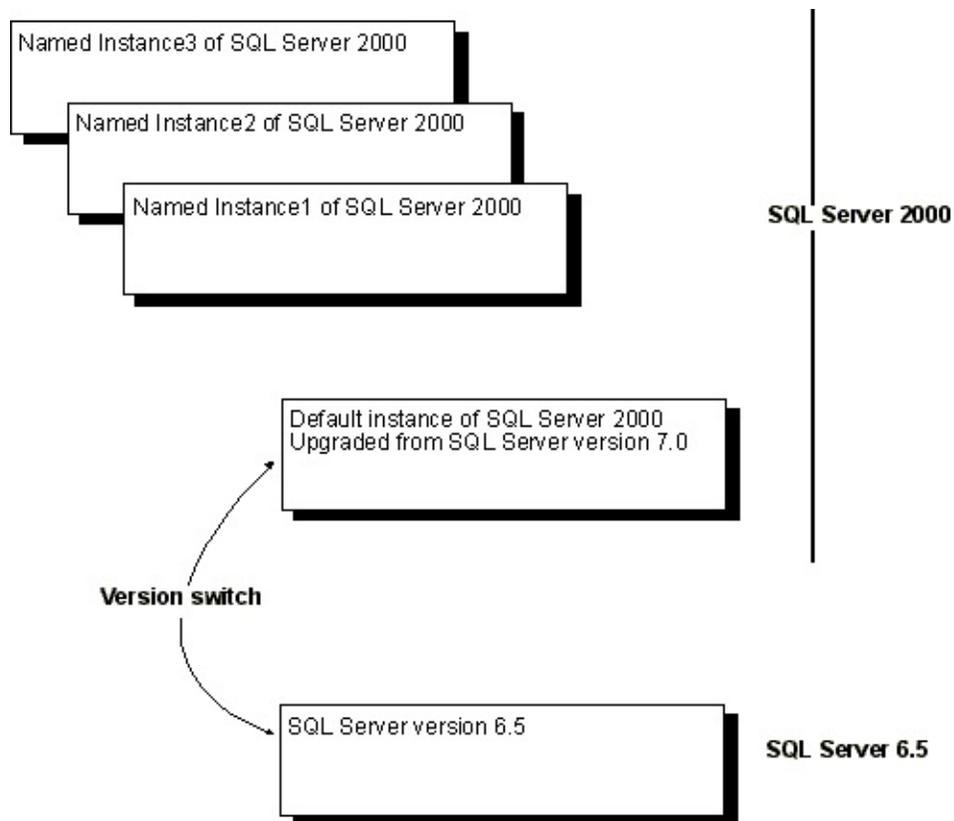
The illustration shows how named instances of SQL Server 2000 and the existing installation of SQL Server version 7.0 can run at the same time. SQL Server version 6.5 is available to be switched in as the default instance instead of SQL Server 7.0.



To use version switching with SQL Server 6.5 and SQL Server 2000, after upgrading from SQL Server version 7.0:

- Keep the SQL Server 6.5 configuration intact.
- Have SQL Server 7.0 installed, but prepare to upgrade SQL Server 7.0 to SQL Server 2000.
- Run Setup. When SQL Server 7.0 is detected, upgrade SQL Server 7.0 to the default instance of SQL Server 2000. (Select the option to Upgrade in the **Existing Installation Options** dialog box, and leave the Default check box selected in the **Instance Name** dialog box.) At this point, the installation of SQL Server 7.0 no longer exists; it is replaced by the default instance of SQL Server 2000.

The illustration shows this configuration, along with three SQL Server 2000 named instances.



## See Also

[Upgrading from SQL Server 7.0 to SQL Server 2000](#)

[Using SQL Server 6.5 with SQL Server 2000](#)

[Running SQL Server 7.0 Along with a Named Instance of SQL Server 2000](#)

# Installing SQL Server

# Collation Options for International Support

In Microsoft® SQL Server™ 2000, it is not required to separately specify code page and sort order for character data, and the collation used for Unicode data. Instead, specify the collation name and sorting rules to use. The term, collation, refers to a set of rules that determine how data is sorted and compared. Character data is sorted using rules that define the correct character sequence, with options for specifying case-sensitivity, accent marks, kana character types, and character width. Microsoft SQL Server 2000 collations include these groupings:

- Windows collations

Windows collations define rules for storing character data based on the rules defined for an associated Windows locale. The base Windows collation rules specify which alphabet or language is used when dictionary sorting is applied, as well as the code page used to store non-Unicode character data. For more information, see [Collations](#).

- SQL collations

SQL collations are provided for compatibility with sort orders in earlier versions of Microsoft SQL Server. For more information, see [Using SQL Collations](#).

## Changing Collations After Setup

When you set up SQL Server 2000, it is important to use the correct collation settings. You can change collation settings after running Setup, but you must rebuild the databases and reload the data. It is recommended that you develop a standard within your organization for these options. Many server-to-server activities can fail if the collation settings are not consistent across servers.

## See Also

[Collation Settings in Setup](#)

[How to rebuild the \*\*master\*\* database \(Rebuild Master utility\)](#)

[Selecting a SQL Collation](#)

[Windows Collation Designators](#)

# Installing SQL Server

## Collation Settings in Setup

Use the **Collation Settings** screen to modify default collation settings. Use the **Windows Locale** option to match collation settings in instances of Microsoft® SQL Server™ 2000. Use **SQL Collations** to match settings that are compatible with the sort orders in earlier versions of SQL Server.

### Windows Locale

Change the default settings for **Windows Locale** (Windows collation) only if your installation of SQL Server must match the collation settings used by another instance of SQL Server 2000, or must match the Windows locale of another computer.

### Collation Designator

Select the name of a specific Windows collation from the list, for example:

- Use **Latin1\_General** for the U.S. English character set (code page 1252).
- Use **Modern\_Spanish** for all variations of Spanish, which also use the same character set as U.S. English (code page 1252).
- Use **Arabic** for all variations of Arabic, which use the Arabic character set (code page 1256).
- Use **Japanese\_Unicode** for the Unicode version of Japanese (code page 932), which has a different sort order from **Japanese**, but the same code page (932).

For more information, see [Windows Collation Designators](#).

### Sort Order

Select Sort Order options to use with the Collation Designator selected. Binary is the fastest sorting order, and is case-sensitive. If **Binary** is selected, the **Case-sensitive**, **Accent-sensitive**, **Kana-sensitive**, and **Width-sensitive** options are not available. For more information, see [Windows Collation Sorting Styles](#).

## **SQL Collations**

The **SQL Collations** option is used for compatibility with earlier versions of Microsoft SQL Server. Select this option to match settings compatible with SQL Server version 7.0, SQL Server version 6.5, or earlier. For more information, see [SQL Collations](#).

# Installing SQL Server

## Windows Collation Sorting Styles

On the **Collation Settings** screen you can choose **Binary** sort order, or you can define the sorting styles to use with the Collation Designator (Windows collation name) selected.

**Note** For Windows collations, the **nchar**, **nvarchar**, and **ntext** data types have the same sorting behavior as **char**, **varchar**, and **text** data types. For more information, see [SQL Server Collation Fundamentals](#).

Sort order	Description
<b>Binary</b>	<p>Sorts and compares data in Microsoft® SQL Server™ tables based on the bit patterns defined for each character. Binary sort order is case-sensitive, that is lowercase precedes uppercase, and accent-sensitive. This is the fastest sorting order.</p> <p>If this option is not selected, SQL Server follows sorting and comparison rules as defined in dictionaries for the associated language or alphabet.</p>
<b>Case-sensitive</b>	<p>Specifies that SQL Server distinguish between uppercase and lowercase letters.</p> <p>If not selected, SQL Server considers the uppercase and lowercase versions of letters to be equal. SQL Server does not define whether lowercase letters sort lower or higher in relation to uppercase letters when Case-sensitive is not selected.</p>
<b>Accent-sensitive</b>	<p>Specifies that SQL Server distinguish between accented and unaccented characters. For example, 'a' is not equal to 'á'.</p> <p>If not selected, SQL Server considers the accented and unaccented versions of letters to be equal.</p>
<b>Kana-sensitive</b>	<p>Specifies that SQL Server distinguish between the two types of Japanese kana characters: Hiragana and</p>

	<p>Katakana.</p> <p>If not selected, SQL Server considers Hiragana and Katakana characters to be equal.</p>
<b>Width-sensitive</b>	<p>Specifies that SQL Server distinguish between a single-byte character (half-width) and the same character when represented as a double-byte character (full-width).</p> <p>If not selected, SQL Server considers the single-byte and double-byte representation of the same character to be equal.</p>

## See Also

[Collation Settings in Setup](#)

[Windows Collation Designators](#)

# Installing SQL Server

## Windows Collation Designators

Use this table to synchronize collation settings with another Windows locale.

In Control Panel, find the Windows locale name in the Regional Settings application (Microsoft® Windows NT® 4.0, Microsoft Windows® 98, and Microsoft Windows 95) or the Regional Options application (Microsoft Windows 2000), and then use this table to find the corresponding Collation Designator and code page.

<b>Windows locale</b>	<b>LCID (locale ID)</b>	<b>Collation designator</b>	<b>Code page</b>
Afrikaans	0xx436	Latin1_General	1252
Albanian	0x41C	Albanian	1250
Arabic (Saudi Arabia)	0x401	Arabic	1256
Arabic (Iraq)	0x801	Arabic	1256
Arabic (Egypt)	0xC01	Arabic	1256
Arabic (Libya)	0x1001	Arabic	1256
Arabic (Algeria)	0x1401	Arabic	1256
Arabic (Morocco)	0x1801	Arabic	1256
Arabic (Tunisia)	0x1C01	Arabic	1256
Arabic (Oman)	0x2001	Arabic	1256
Arabic (Yemen)	0x2401	Arabic	1256
Arabic (Syria)	0x2801	Arabic	1256
Arabic (Jordan)	0x2C01	Arabic	1256
Arabic (Lebanon)	0x3001	Arabic	1256
Arabic (Kuwait)	0x3401	Arabic	1256
Arabic (United Arab Emirates)	0x3801	Arabic	1256
Arabic (Bahrain)	0x3C01	Arabic	1256
Arabic (Qatar)	0x4001	Arabic	1256
Basque	0x42D	Latin1_General	1252
Byelorussian	0x423	Cyrillic_General	1251
Bulgarian	0x402	Cyrillic_General	1251

Catalan	0x403	Latin1_General	1252
Chinese (Taiwan)	0x30404	Chinese_Taiwan_Bopomofo	950
Chinese (Taiwan)	0x404	Chinese_Taiwan_Stroke	950
Chinese (People's Republic of China)	0x804	Chinese_PRC	936
Chinese (People's Republic of China)	0x20804	Chinese_PRC_Stroke	936
Chinese (Singapore)	0x1004	Chinese_PRC	936
Croatia	0x41a	Croatian	1250
Czech	0x405	Czech	1250
Danish	0x406	Danish_Norwegian	1252
Dutch (Standard)	0x413	Latin1_General	1252
Dutch (Belgium)	0x813	Latin1_General	1252
English (United States)	0x409	Latin1_General	1252
English (Britain)	0x809	Latin1_General	1252
English (Canada)	0x1009	Latin1_General	1252
English (New Zealand)	0x1409	Latin1_General	1252
English (Australia)	0xC09	Latin1_General	1252
English (Ireland)	0x1809	Latin1_General	1252
English (South Africa)	0x1C09	Latin1_General	1252
English (Caribbean)	0x2409	Latin1_General	1252
English (Jamaican)	0x2009	Latin1_General	1252
Estonian	0x425	Estonian	1257
Faeroese	0x0438	Latin1_General	1252
Farsi	0x429	Arabic	1256
Finnish	0x40B	Finnish_Swedish	1252
French (Standard)	0x40C	French	1252
French (Belgium)	0x80C	French	1252
French (Switzerland)	0x100C	French	1252
French (Canada)	0xC0C	French	1252
French (Luxembourg)	0x140C	French	1252
Georgian (Modern)	0x10437	Georgian_Modern_Sort	1252

Sort)			
German (PhoneBook Sort)	0x10407	German_PhoneBook	1252
German (Standard)	0x407	Latin1_General	1252
German (Switzerland)	0x807	Latin1_General	1252
German (Austria)	0xC07	Latin1_General	1252
German (Luxembourg)	0x1007	Latin1_General	1252
German (Liechtenstein)	0x1407	Latin1_General	1252
Greek	0x408	Greek	1253
Hebrew	0x40D	Hebrew	1255
Hindi	0x439	Hindi	Unicode only
Hungarian	0x40E	Hungarian	1250
Hungarian	0x104E	Hungarian_Technical	1250
Icelandic	0x40F	Icelandic	1252
Indonesian	0x421	Latin1_General	1252
Italian	0x410	Latin1_General	1252
Italian (Switzerland)	0x810	Latin1_General	1252
Japanese	0x411	Japanese	932
Japanese (Unicode)	0x10411	Japanese_Unicode	932
Korean (Extended Wansung)	0x412	Korean_Wansung	949
Korean	0x412	Korean_Wansung_Unicode	949
Latvian	0x426	Latvian	1257
Lithuanian	0x427	Lithuanian	1257
Lithuanian	0x827	Lithuanian_Classic	1257
Macedonian	0x41C	Cyrillic_General	1251
Norwegian (Bokmål)	0x414	Danish_Norwegian	1252
Norwegian (Nynorsk)	0x814	Danish_Norwegian	1252
Polish	0x415	Polish	1250
Portuguese (Standard)	0x816	Latin1_General	1252
Portuguese (Brazil)	0x416	Latin1_General	1252

Romanian	0x418	Romanian	1250
Russian	0x419	Cyrillic_General	1251
Serbian (Latin)	0x81A	Cyrillic_General	1251
Serbian (Cyrillic)	0xC1A	Cyrillic_General	1251
Slovak	0x41B	Slovak	1250
Slovenian	0x424	Slovenian	1250
Spanish (Mexico)	0x80A	Traditional_Spanish	1252
Spanish (Traditional Sort)	0x40A	Traditional_Spanish	1252
Spanish (Modern Sort)	0xC0A	Modern_Spanish	1252
Spanish (Guatemala)	0x100A	Modern_Spanish	1252
Spanish (Costa Rica)	0x140A	Modern_Spanish	1252
Spanish (Panama)	0x180A	Modern_Spanish	1252
Spanish (Dominican Republic)	0x1C0A	Modern_Spanish	1252
Spanish (Venezuela)	0x200A	Modern_Spanish	1252
Spanish (Colombia)	0x240A	Modern_Spanish	1252
Spanish (Peru)	0x280A	Modern_Spanish	1252
Spanish (Argentina)	0x2C0A	Modern_Spanish	1252
Spanish (Ecuador)	0x300A	Modern_Spanish	1252
Spanish (Chile)	0x340A	Modern_Spanish	1252
Spanish (Uruguay)	0x380A	Modern_Spanish	1252
Spanish (Paraguay)	0x3C0A	Modern_Spanish	1252
Spanish (Bolivia)	0x400A	Modern_Spanish	1252
Swedish	0x41D	Finnish_Swedish	1252
Thai	0x41E	Thai	874
Turkish	0x41F	Turkish	1254
Ukrainian	0x422	Ukrainian	1251
Urdu	0x420	Arabic	1256
Vietnamese	0x42A	Vietnamese	1258

**See Also**

[Collation Settings in Setup](#)

[Collations](#)

[Windows Collation Sorting Styles](#)

[Windows Collation Name](#)

# Installing SQL Server

## Using SQL Collations

SQL collation settings correspond to the type of installation. In general, choose a SQL collation that supports the Windows locale most commonly used at your site. For more information about identifying your site Windows Locale, see Regional Settings in Windows Control Panel. In many cases, a computer will run the Windows locale that matches the language requirements of the user, so Setup automatically detects the Windows locale and chooses the appropriate SQL collation.

SQL collations control:

- The code page used for storing non-Unicode data in Microsoft® SQL Server™.
- The rules governing how SQL Server sorts and compares characters stored in both Unicode and non-Unicode data types.

Choose a SQL collation if:

- You use the replication feature with existing instances of SQL Server version 6.5 or SQL Server version 7.0
- Your application code depends on the behaviors of the previous SQL Server collations.

An upgrade of SQL Server 7.0 to SQL Server 2000 keeps the previous SQL collation settings; no collation choice is required.

Use this table to determine if you need to make a collation choice, and if so, which collation you should choose.

<b>Installation you want</b>	<b>Collation to choose</b>
To install on a new system with no compatibility requirements for synchronizing with any type of existing system	Use the locale identified by Setup, and then choose the desired binary, case, or other options.

	<p>For this release of SQL Server, when Setup detects that the computer is running the U.S. English locale, Setup automatically selects the SQL collation: <b>Dictionary order, case-insensitive, for use with 1252 character set.</b></p> <p>To select the equivalent Windows collation, select <b>Collation designator</b>, choose the <b>Latin1_General</b> collation designator, do not select <b>case-sensitive</b>, and select <b>accent-sensitive</b>.</p>
<p>To upgrade an installation of SQL Server 6.5 or SQL Server 7.0 to a default instance of SQL Server 2000, or to install a default instance of SQL Server 2000 that will version switch with an installation of SQL Server 6.5</p>	<p>Use the SQL collation chosen by Setup.</p>
<p>To synchronize (for example, to replicate) with an existing instance of SQL Server 2000</p>	<p>Select <b>SERVERPROPERTY(N'Collation')</b> on the existing instance, and specify that collation. If the collation name of the existing instance starts with SQL, select the same SQL collation in Setup. If the collation name of the existing instance does not start with SQL, the collation name refers to a Windows collation name and consists of the collation designator name followed by a description of what binary, case, accent, kana and width sensitivity options are specified. Select the same Windows collation designator and sorting options in</p>

	Setup.
To synchronize with an existing installation of SQL Server 6.5 or SQL Server 7.0	<p>Execute <b>sp_helpsort</b> on the existing system, and then use the sort ID to select a SQL collation to make your instance of SQL Server 2000 compatible with an existing installation.</p> <p>For more information, see <a href="#">Selecting a SQL Collation</a>.</p>
To synchronize with a Windows locale of another computer	<p>In Control Panel, find the locale name from the Regional Settings application (Microsoft Windows NT® 4.0, Microsoft Windows® 98, and Microsoft Windows 95), or from the Regional Options application (Microsoft Windows 2000), and then use the table provided in the topic <a href="#">Windows Collation Designators</a>. Set the sorting options, as explained in the topic <a href="#">Windows Collation Sorting Styles</a>.</p>

**Note** When you perform an action that depends on collations, the SQL Server collation used by the referenced object must use a code page supported by the operating system running on the computer. For more information, see [Specifying Collations](#).

## See Also

[Examples of SQL Collations](#)

[Selecting Collations](#)

[sp\\_helpsort](#)

[Setting Client Code Pages](#)

[SQL Server Collation Fundamentals](#)

[SERVERPROPERTY](#)

## Installing SQL Server

## Examples of SQL Collations

These are examples of SQL collations listed on the **Collation Settings** screen in Microsoft® SQL Server™ 2000 Setup:

### **Binary order, for use with the 437 (U.S. English) character set.**

This collation uses binary sort order (simple sorting based on coded value) with the U.S. English character set (code page 437 - MS-DOS Latin US).

In Transact-SQL, the string `SQL_Latin1_General_Cp437_BIN` is used to designate this setting.

### **Dictionary order, case-insensitive, accent-insensitive, for use with 1252 character set.**

This collation uses the dictionary sorting rules for the U.S. English character set (code page 1252 - Windows Latin 1 ANSI, sort order ID 54). Uppercase or lowercase characters and accent marks are not considered when sorting.

In Transact-SQL, the string `SQL_Latin1_General_CP1_CI_AI` is used to designate this setting.

### **Romanian dictionary order, case-sensitive, for use with the 1250 (Central European) character set.**

This collation uses the dictionary order sorting rules for the Romanian language, and uses the Central European character set (code page 1250, sort order ID 89).

In Transact-SQL, the string `SQL_Romanian_Cp1250_CS_AS` is used to designate this setting.

## See Also

[Collation Settings in Setup](#)

[Selecting a SQL Collation](#)

[Using SQL Collations](#)

[SQL Collation Name](#)

# Installing SQL Server

## Selecting a SQL Collation

When selecting a SQL collation in the **Collations Settings** screen, use the following table to make the installation of Microsoft® SQL Server™ 2000 compatible with an installation of an earlier version of SQL Server.

**Note** Each SQL Collation name has an equivalent string in T-SQL code. For a list of sort order identifiers and the T-SQL version, see [SQL Collation Name](#).

In the table, the left column lists the sort order ID of an instance of SQL Server 7.0 or SQL Server 6.5. The right column lists the SQL Server 2000 collation recommended for compatibility.

Sort order ID	SQL collation name
30	Binary order, for use with the 437 (U.S. English) character set.
31	Dictionary order, case-sensitive, for use with the 437 (U.S. English) character set.
32	Dictionary order, case-insensitive, for use with the 437 (U.S. English) character set.
33	Dictionary order, case-insensitive, uppercase preference, for use with the 437 (U.S. English) character set.
34	Dictionary order, case-insensitive, accent-insensitive, for use with the 437 (U.S. English) character set.
40	Binary order, for use with the 850 (Multilingual) character set.
41	Dictionary order, case-sensitive, for use with the 850 (Multilingual) character set.
42	Dictionary order, case-insensitive, for use with the 850 (Multilingual) character set.
43	Dictionary order, case-insensitive, uppercase preference, for use with the 850 (Multilingual) character set.
44	Dictionary order, case-insensitive, accent-insensitive, for use with the 850 (Multilingual) character set.
49	Strict compatibility with version 1.x case-insensitive databases, for use with the 850 (Multilingual) character set.

50	Binary order for use with 1252 character set.
51	Dictionary order, case-sensitive, for use with 1252 character set.
52	Dictionary order, case-insensitive, for use with 1252 character set.
53	Dictionary order, case-insensitive, uppercase preference, for use with 1252 character set.
54	Dictionary order, case-insensitive, accent-insensitive, for use with 1252 character set.
55	Alternate dictionary order, case-sensitive, for use with the 850 (Multilingual) character set.
56	Alternate dictionary order, case-insensitive, uppercase preference, for use with the 850 (Multilingual) character set.
57	Alternate dictionary order, case-insensitive, accent-insensitive, for use with the 850 (Multilingual) character set.
58	Scandinavian dictionary order, case-insensitive, uppercase preference, for use with the 850 (Multilingual) character set.
59	Scandinavian dictionary order, case-sensitive, for use with the 850 (Multilingual) character set.
60	Scandinavian dictionary order, case-insensitive, for use with the 850 (Multilingual) character set.
61	Alternate dictionary order, case-insensitive, for use with the 850 (Multilingual) character set.
71	Latin-1 case-sensitive, for use with 1252 character set.
72	Latin-1 case-insensitive, for use with 1252 character set.
73	Danish/Norwegian case-sensitive sort order for code page 1252.
74	Finnish/Swedish case-sensitive sort order for code page 1252.
75	Icelandic case-sensitive sort order for code page 1252.
80	Binary order, for use with the 1250 (Central European) character set.
81	Dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
82	Dictionary order, case-insensitive, for use with the 1250

	(Central European) character set.
83	Czech dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
84	Czech dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
85	Hungarian dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
86	Hungarian dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
87	Polish dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
88	Polish dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
89	Romanian dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
90	Romanian dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
91	Croatian dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
92	Croatian dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
93	Slovak dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
94	Slovak dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
95	Slovenian dictionary order, case-sensitive, for use with the 1250 (Central European) character set.
96	Slovenian dictionary order, case-insensitive, for use with the 1250 (Central European) character set.
97	Windows Polish case-sensitive sort order for code page 1250.
98	Windows Polish case-insensitive sort order for code page 1250.
104	Binary order, for use with the 1251 (Cyrillic) character set.
105	Dictionary order, case-sensitive, for use with the 1251 (Cyrillic) character set.

106	Dictionary order, case-insensitive, for use with the 1251 (Cyrillic) character set.
107	Ukrainian dictionary order, case-sensitive, for use with the 1251 (Cyrillic) character set.
108	Ukrainian dictionary order, case-insensitive, for use with the 1251 (Cyrillic) character set.
112	Binary order, for use with the 1253 (Greek) character set.
113	Dictionary order, case-sensitive, for use with the 1253 (Greek) character set.
114	Dictionary order, case-insensitive, for use with the 1253 (Greek) character set.
120	Mixed dictionary order, for use with the 1253 (Greek) character set.
121	Dictionary order, case-sensitive, accent-sensitive, for use with the 1253 (Greek) character set.
124	Dictionary order, case-insensitive, accent-insensitive, for use with the 1253 (Greek) character set.
128	Binary order, for use with the 1254 (Turkish) character set.
129	Dictionary order, case-sensitive, for use with the 1254 (Turkish) character set.
130	Dictionary order, case-insensitive, for use with the 1254 (Turkish) character set.
136	Binary order, for use with the 1255 (Hebrew) character set.
137	Dictionary order, case-sensitive, for use with the 1255 (Hebrew) character set.
138	Dictionary order, case-insensitive, for use with the 1255 (Hebrew) character set.
144	Binary order, for use with the 1256 (Arabic) character set.
145	Dictionary order, case-sensitive, for use with the 1256 (Arabic) character set.
146	Dictionary order, case-insensitive, for use with the 1256 (Arabic) character set.
152	Binary order, for use with the 1257 (Baltic) character set.
153	Dictionary order, case-sensitive, for use with the 1257 (Baltic) character set.

154	Dictionary order, case-insensitive, for use with the 1257 (Baltic) character set.
155	Estonian dictionary order, case-sensitive, for use with the 1257 (Baltic) character set.
156	Estonian dictionary order, case-insensitive, for use with the 1257 (Baltic) character set.
157	Latvian dictionary order, case-sensitive, for use with the 1257 (Baltic) character set.
158	Latvian dictionary order, case-insensitive, for use with the 1257 (Baltic) character set.
159	Lithuanian dictionary order, case-sensitive, for use with the 1257 (Baltic) character set.
160	Lithuanian dictionary order, case-insensitive, for use with the 1257 (Baltic) character set.
183	Danish/Norwegian dictionary order, case-insensitive, uppercase preference, for use with 1252 character set.
184	Swedish/Finnish (Standard) dictionary order, case-insensitive, uppercase preference, for use with 1252 character set.
185	Swedish/Finnish (Phone) dictionary order, case-insensitive, uppercase preference, for use with 1252 character set.
186	Icelandic dictionary order, case-insensitive, uppercase preference, for use with 1252 character set.
192	Binary order, for use with the 932 (Japanese) character set.
193	Dictionary order, case-insensitive, for use with the 932 (Japanese) character set
194	Binary order, for use with the 949 (Korean) character set.
195	Dictionary order, case-insensitive, for use with the 949 (Korean) character set.
196	Binary order, for use with the 950 (Traditional Chinese) character set.
197	Dictionary order, case-insensitive, for use with the 950 (Traditional Chinese) character set.
198	Binary order, for use with the 936 (Simplified Chinese) character set.

199	Dictionary order, case-insensitive, for use with the 936 (Simplified Chinese) character set.
200	Dictionary order, case-sensitive, for use with the 932 (Japanese) character set.
201	Dictionary order, case-sensitive, for use with the 949 (Korean) character set.
202	Dictionary order, case-sensitive, for use with the 950 (Traditional Chinese) character set.
203	Dictionary order, case-sensitive, for use with the 936 (Simplified Chinese) character set.
204	Binary order, for use with the 874 (Thai) character set.
205	Dictionary order, case-insensitive, for use with the 874 (Thai) character set.
206	Dictionary order, case-sensitive, for use with the 874 (Thai) character set.

## See Also

[Examples of SQL Collations](#)

[Collation Settings in Setup](#)

[Specifying Collations](#)

[SQL Collations](#)

# Installing SQL Server

## **Setting Client Code Pages**

The code pages a client uses are determined by your operating system settings.

**To set client code pages in the Windows NT, Windows 2000, or Windows 98 operating systems**

## Installing SQL Server

# Upgrading Character Set, Sort Order, and Collation

Microsoft® SQL Server™ 2000 supports several different ways to specify collations. You no longer have to separately specify the code page used for character data, the sort order used for character data, and the collation used for Unicode data. When you upgrade, SQL collations can be specified for compatibility with existing instances of SQL Server.

Because the default collation for an instance of Microsoft SQL Server is defined during setup, it is important to become familiar with collation settings in SQL Server 2000 when:

- Your application code depends in some way on the behavior of previous SQL Server collations.
- You are going to use the replication feature with existing installations of SQL Server 6.5 or SQL Server 7.0.
- You must store character data that reflects multiple languages.

## See Also

[Collation Options for International Support](#)

[Collations](#)

[Selecting Collations](#)

[Specifying the Default Collation for an Instance of SQL Server](#)

# Installing SQL Server

## Changing Collation Settings After Installing

Collation settings, which include character set, sort order, and other locale-specific settings, are fundamental to the structure of all Microsoft® SQL Server™ 2000 databases. To change one or more of these settings, you must rebuild the **master** and user databases.

### See Also

[Collation Settings in Setup](#)

[Collations](#)

[How to rebuild the \*\*master\*\* database \(Rebuild Master utility\)](#)

# Installing SQL Server

## Upgrading to SQL Server 2000: Overview

Upgrading from Microsoft® SQL Server™ version 7.0 to Microsoft SQL Server 2000 is one of the basic choices offered by the SQL Server Setup program on the initial **Installation Selection** screen. When you select the option to **Upgrade, remove, or add components to an existing installation of SQL Server**, Setup detects your current installation and initiates the correct sequence of setup screens for the upgrade selected. Upgrade variations include:

- A complete installation upgrade from SQL Server 7.0 to SQL Server 2000 (installing over SQL Server 7.0).
- Adding components to an installation of SQL Server 2000.
- An upgrade to the feature set of an existing installation of SQL Server 2000 (edition and component upgrade).
- An upgrade to SQL Server 2000 from SQL Server version 6.5 using the SQL Server Upgrade Wizard.
- An online database upgrade of SQL Server 7.0 databases to SQL Server 2000 database format using the Copy Database Wizard.

During the upgrade from SQL Server 7.0, external packages, such as Microsoft Management Console and the Microsoft Distributed Transaction Coordinator, must be installed for each upgrade, and the registry updated. The **master** database and other system databases are upgraded in various ways involving a series of scripts run on the server with specific options. If the upgrade process fails built-in recovery mechanisms restart and resume the upgrade.

### See Also

[Upgrading from SQL Server 7.0 to SQL Server 2000](#)

[Upgrading an Existing Installation of SQL Server](#)

[Upgrading Databases from SQL Server 7.0 \(Copy Database Wizard\)](#)

[Upgrading to a SQL Server 2000 Failover Cluster](#)

## Installing SQL Server

## Hardware and Software Requirements for Upgrading

In addition to the hardware and software requirements for an installation of Microsoft® SQL Server™, the computer must meet these requirements for an upgrade.

Hardware/software	Upgrade requirements
Operating system	Microsoft Windows NT® Server Enterprise Edition version 4.0 with Service Pack 5 (SP5) or later.  Windows NT Server version 4.0 with SP5 or later.  Windows NT Workstation 4.0 with SP5 or later.  Internet Explorer 5.0 or later.  Windows 2000.
SQL Server 6.5	When upgrading SQL Server version 6.5 to an instance of SQL Server 2000 on the same computer, you must have applied SQL Server 6.5 Service Pack 5 (SP5) or later. When upgrading SQL Server 6.5 to an instance of SQL Server 2000 on a different computer, you must have applied SQL Server 6.5 Service Pack 3 (SP3) or later.
SQL Server 7.0	SQL Server 7.0 (at any Service Pack level).
Network protocols	Named Pipes.  SQL Server 6.5, SQL Server 7.0, and SQL Server 2000 all must be set to listen to the default pipe, \\.\pipe\sql\query. Named Pipes is required even for a tape backup upgrade.
Hard-disk space	No additional hard-disk space is required when upgrading from SQL Server 7.0 to SQL Server 2000.  When upgrading from SQL Server 6.5 to SQL Server 2000, however, you need approximately 1.5 times the size of the SQL Server 6.5 databases.

## **See Also**

[Hardware and Software Requirements for Installing SQL Server 2000](#)

# Installing SQL Server

## Upgrading from SQL Server 7.0 to SQL Server 2000

You can overwrite an installation of Microsoft® SQL Server™ version 7.0 with a version upgrade to Microsoft SQL Server 2000. If SQL Server 7.0 is detected as an existing installation when you run Setup, you can choose the option to upgrade. In this process, all the SQL Server 7.0 program files are upgraded, and all data stored in SQL Server 7.0 databases is preserved. In addition, SQL Server Books Online for SQL Server 7.0 remains on your computer.

**Note** SQL Server 7.0 profiler traces and registered servers are not upgraded when SQL Server 7.0 tools are upgraded to SQL Server 2000. Similarly, information models that were installed with Microsoft Repository 2.0 are not upgraded automatically. SQL Server 2000 supports newer versions of information models for both Data Transformation Services (DTS) and the Open Information Model (OIM). For more information about upgrading the DTS information model, see [DTS Information Model](#). For more information about upgrading the OIM, see [Upgrading an Information Model](#).

You can also upgrade from one edition of SQL Server to another edition during the version upgrade to SQL Server 2000. For more information, see [SQL Server 2000: Editions and Components](#).

**CAUTION** After you perform this version upgrade, the SQL Server 7.0 installation no longer exists on your computer. The only way to restore an installation of SQL Server 7.0 is to first uninstall SQL Server 2000, perform a complete reinstall of SQL Server 7.0 files, and then restore your backed-up SQL Server 7.0 databases.

**To upgrade an installation of SQL Server 7.0 to SQL Server 2000**

## Installing SQL Server

## Replication and Upgrading

When upgrading to Microsoft® SQL Server™ 2000, you can upgrade servers in your organization one at a time; however, when servers are used for replication, you must upgrade the Distributor first, the Publisher second, and then Subscribers. Upgrading servers one at a time following this sequence is recommended when a large number of Publishers and Subscribers exist because you can continue to replicate data even though servers are running different versions of SQL Server. You can create new publications and subscriptions with servers running instances of SQL Server 2000, and still maintain subscriptions created in SQL Server 6.5 or SQL Server 7.0.

When using transactional replication, you can upgrade Subscribers before the Publisher. If you are using immediate updating with snapshot replication or transactional replication, there are additional upgrade recommendations in this topic under Upgrading and Immediate Updating.

You can upgrade replication servers running SQL Server 6.5 or SQL Server 7.0 to SQL Server 2000. If the server is running SQL Server 6.5, you do not need to upgrade it to SQL Server 7.0 before upgrading to SQL Server 2000.

**IMPORTANT** When upgrading servers configured for replication to SQL Server 2000, the database compatibility level must be set to 70 (version 7.0 compatibility) or later. If you have servers running in 65 (version 6.5) or an earlier compatibility level, temporarily change them to 70 or later during the upgrade process.

When the Publisher or Subscriber is running in 65 or an earlier compatibility level during upgrade to SQL Server 2000, error 15048 will be raised stating that the operation is supported only on SQL Server version 7.0 or SQL Server 2000.

For more information about setting the backward compatibility level, see [SQL Server 2000 and SQL Server version 6.5](#).

If you are upgrading replication on a failover cluster, you must uncluster the previous installation before upgrading. Unclustering the previous installation means that you must delete all publications, remove replication, and reconfigure it after upgrading to SQL Server 2000. This will not be a requirement when

upgrading SQL Server 2000 to future releases.

## Upgrading and Immediate Updating

If you are using immediate updating with snapshot replication or transactional replication, changes to that feature in SQL Server 2000 will affect how you upgrade. Rows in immediate updating articles now use a **uniqueidentifier** column to identify versions, whereas in SQL Server 7.0, a **timestamp** column was used. In addition, the triggers generated for immediate updating have been changed, and the trigger generation code has been modified to accommodate queued updating. Because of these changes, additional upgrade steps are necessary.

If using immediate updating:

- Upgrade both the Publisher and Subscriber before replicating data.
- Drop the publication and all subscriptions to the publication.
- Use an ALTER TABLE DROP COLUMN Transact-SQL statement to drop the **timestamp** column from the tables on the Publisher and from the tables on the Subscriber that allow Subscriber updates.
- Re-create the publication and subscriptions. The system adds a **uniqueidentifier** column to the published table. That column is used for row versioning (to detect conflicts when receiving updates from the Subscriber).

Although it is recommended you upgrade both the Publisher and the Subscriber and then drop and re-create the existing publications, the Publisher and Subscribers can be upgraded in any order. If you need to reinitialize a Subscriber or add a new Subscriber, you need to drop and re-create the publication.

## Upgrading and File Transfer Protocol

If using File Transfer Protocol (FTP), you should follow the recommended upgrade path, which ensures that Subscribers are able to obtain the necessary

FTP information from the Distributor.

SQL Server 2000 stores FTP parameters as Publication Properties; you no longer need to administer them at the Subscriber for each subscription. When upgrading to SQL Server 2000, the FTP option in the Publication Properties is turned off, and you need to open the properties for each publication that uses FTP, and then reset the FTP parameters.

SQL Server 7.0 Subscribers will continue to locate FTP files using the FTP parameters stored in the Subscription Properties when using a Distributor running an instance of SQL Server 2000. However, Subscribers running an instance of SQL Server 2000 will not be able to obtain FTP information from Distributors running earlier versions of SQL Server.

Existing subscriptions using merge replication or transactional replication will be unaffected by this change unless you need to reinitialize or connect to the FTP site. The FTP parameters need to be specified before snapshot replication occurs, or replication agents will not be able to locate the snapshot files.

For more information about changing the FTP parameters, see [Using TCP/IP and FTP](#) and [How to specify FTP information \(Enterprise Manager\)](#).

## **Troubleshooting and Replication Upgrades**

If errors occur while upgrading replication servers, they might be related to the database being offline or unavailable or a script may have failed. For more information about troubleshooting errors that occur when upgrading replication, see [Help with Replication](#).

It is recommended that you stop all data modifications at the replication server while it is being upgraded. When upgrading from SQL Server 6.5, you must run the Log Reader Agent and Distribution Agent before upgrading to make sure there are no replicated commands pending delivery to Subscribers.

Because you can upgrade servers running instances of Microsoft® SQL Server™ 2000 one at a time, you may have circumstances where servers in your replication topology are running different versions of SQL Server. You can replicate between different versions of SQL Server, but you are often limited to the functionality of the earliest version used.

**IMPORTANT** When upgrading from SQL Server 6.5 or 7.0 to SQL Server 2000,

SQL Server Setup runs several \*.sql replication scripts. Although the upgrade process can take several minutes and does not display progress notifications, you can view error messages in the \*.out and \*.err files located in the SQL Server Install directory.

## **See Also**

[Publishing Data Over the Internet Using TCP/IP and FTP](#)

[Replication Between Different Versions of SQL Server](#)

[Replication Data Considerations](#)

[Updatable Subscriptions](#)

## Installing SQL Server

## **Upgrading Databases from SQL Server 7.0 (Copy Database Wizard)**

As an enhancement to the regular upgrade procedure, you can perform an online upgrade of databases and associated meta data. Using the Copy Database Wizard, you can move or copy a database from Microsoft® SQL Server™ 7.0 to an instance of Microsoft SQL Server 2000, without having to shut down any servers in the process.

Advantages of an online database upgrade include:

- No downtime for servers during the upgrade.
- Custom selection of databases to upgrade, leaving other databases still available to the original (SQL Server 7.0) server.
- Inclusion of related meta data in the upgrade procedure. For example, logon information, jobs, and user-specific objects associated with user databases can be included.
- The process can be run at a convenient time.

The Database Copy Wizard is based on detach and attach functionality that allows user databases to be moved or copied from a source to a destination server. A Data Transformation Services (DTS) package performs the actual move or copy operation. You can schedule the package to run at a specified time or rerun the package if required.

### **Options for SQL Server 7.0 Database Upgrades**

Database administrators can move or copy one or more databases from an instance of SQL Server 7.0 to the default instance of SQL Server 2000 on your local computer, or to either a default or a named instance on a remote computer. This upgrade feature does not support SQL Server 6.5 databases.

## Local computer

- SQL Server 7.0 databases can be upgraded to a named instance of SQL Server 2000 on the local computer.

## Remote computer

- SQL Server 7.0 databases can be upgraded to a default instance of SQL Server 2000 on a remote computer.
- SQL Server 7.0 databases can be upgraded to a named instance of SQL Server 2000 on a remote computer.

**Note** You can have only one active default instance of SQL Server on a computer at one time; either a default instance of SQL Server 7.0 or a default instance of SQL Server 2000. SQL Server 6.5 can also be a default instance. For more information, see [Working with Instances and Versions of SQL Server](#).

## Exceptions

The Copy Database Wizard cannot be used in these situations:

- A database with the identical name on both source and destination servers cannot be moved or copied. On the database selection screen, it will be noted as "Already exists."
- For databases involved in replication, a regular server upgrade is required.

## Copy Database Wizard Safeguards

At the start of a database move or copy operation, one administrator must have exclusive use of all files to prevent any changes to the file set during the process. Two connections are required to copy database files: **sysadmin** privileges on both installations of SQL Server and administrator privileges on the server/network.

To prevent any chance of data corruption, the SQL Server 7.0 databases must be

in read-only condition and cannot be renamed during this operation. Any name conflicts between source and destination servers must be resolved manually prior to upgrading databases. Nothing on the destination server is overwritten.

If you move or copy multiple databases in one operation, each database is actually moved one at a time; that is, one database at a time is detached, files are copied and then reattached. To avoid any problems, the DTS package writes a message to the error log indicating that the database is about to be detached from its source server. At the same time, a script is prepared to attach the database to its destination. After the database is successfully attached to the destination, another entry is written to the log indicating successful completion.

When upgrading to a destination that is a clustered server, the Copy Database Wizard will ensure you select only shared drives on a clustered destination server. The source server may also be clustered.

**Note** Unrelated to this upgrade process, you can also use the Copy Database Wizard to move or copy user databases from one instance of SQL Server 2000 to another instance of SQL Server 2000. For more information, see [Using the Copy Database Wizard](#).

**To upgrade databases online using the Copy Database Wizard**

# Installing SQL Server

## Upgrading Databases from SQL Server 6.5 (Upgrade Wizard)

You can convert data from Microsoft® SQL Server™ version 6.5 to the formats for SQL Server 2000 using the SQL Server Upgrade Wizard. The wizard upgrades any or all of your databases, transferring all catalog data, objects, and user data. It also transfers replication settings, SQL Executive settings, and most of the SQL Server 6.5 configuration options. Be sure to review all aspects of this upgrade, as noted in [Preparing to Upgrade from SQL Server 6.5](#).

**Note** To run the SQL Server Upgrade Wizard, you must have a default instance of Microsoft SQL Server 2000 installed on your computer.

The SQL Server Upgrade Wizard does not support consolidation of databases from multiple SQL Server 6.5 installations. If you must upgrade SQL Server 6.5 databases from multiple servers, consolidate all of the SQL Server 6.5 databases onto one server, and then run the wizard to upgrade the consolidated server.

The SQL Server Upgrade Wizard does not remove SQL Server 6.5 from your computer. If you are using a tape backup to perform the upgrade, you have the option of removing the SQL Server 6.5 devices to save disk space.

When the upgrade process is complete, two separate installations of SQL Server exist, including two separate sets of the same data. The SQL Server 6.5 and the SQL Server 2000 installations become independent of each other.

If you are performing the upgrade on a single computer, additional disk space is required. For more information, see [Estimating the Disk Space Required for Upgrading](#). You can also upgrade from one computer to another. For more information, see [Upgrading Using One or Two Computers \(Logon Screen\)](#).

**Note** You can leave the installation of SQL Server 6.5 on a computer indefinitely. In addition to installations of SQL Server 6.5 and a default instance of SQL Server 2000 on the same computer, you also can install multiple named instances of the SQL Server 2000 on the same computer. For more information, see [Working with Instances and Versions of SQL Server](#).

**To perform a version upgrade using a direct pipeline**

# Installing SQL Server

## Preparing to Upgrade from SQL Server 6.5

Follow this checklist before using the SQL Server Upgrade Wizard to move from Microsoft® SQL Server™ version 6.5 to Microsoft SQL Server 2000:

- Back up the SQL Server 6.5 database files (all .dat files, including **master**) so you can completely restore them if necessary.
- Run the appropriate Database Console Commands (DBCC) on the SQL Server 6.5 databases to ensure they are in a consistent state.
- Estimate the disk space required. In addition to the hard disk space used by Microsoft SQL Server 2000, you need approximately 1.5 times the size of the SQL Server 6.5 databases.
- Set **tempdb** to at least 10 MB in the SQL Server 6.5 installation, 25 MB is recommended.
- Ensure the **master** database has at least 3 MB of free space.
- Ensure that all database users have logon information in the **master** database.

This is important for restoring a database because system logon information resides in the **master** database.

- Ensure the @@SERVERNAME is defined on SQL Server 2000. If @@SERVERNAME is NULL, you can use the sp\_addserver system stored procedure. For example, if your computer is named production, the command would be sp\_addserver 'production1',local. Changes do not take affect until the MSSQLServer service is restarted.

**Note** Because SQL Server 6.5 does not recognize the hyphen (-) in a computer name, replace a hyphen with an underscore (\_).

- Disable any startup stored procedures.

The SQL Server Upgrade Wizard starts and stops the SQL Server 6.5 server during the upgrade process. Stored procedures processed at startup may cause the upgrade process to stop responding.

- Ensure that you upgrade all databases with cross-database dependencies at the same time.

For example, you want to upgrade three databases, **database1**, **database2**, and **database4**, and there is logon information in SQL Server 6.5 **master..sysdatabases** for **USER1** that defaults to **database3** (not one of the databases you are upgrading). The SQL Server Upgrade Wizard does not create the logon information because the database is not upgraded, and therefore does not exist in SQL Server 2000. If **USER1** is listed as the owner for objects in any of the databases upgraded, those objects cannot be created because the logon information for **USER1** does not exist.

- If performing a two-computer upgrade, assign a domain user name and password to the MSSQLServer service for SQL Server 6.5 and SQL Server 2000 instead of using the local system account or a local user account. The domain user account should belong to the **Administrators** group of both the computers involved in the upgrade. (The local system account is sufficient for a one-computer upgrade.)
- Stop replication and ensure that the log is empty.
- Quit all applications, including all services dependent on SQL Server.

If you copied the SQL Server 6.5 databases to a new computer to perform the upgrade, you may need to update the new SQL Server 6.5 **master** database as follows:

- Change references from the earlier server name to the current server name in the SQL Server 6.5 **master** database.
- Update the device file locations in the SQL Server 6.5 **master** database.

- Ensure all users have corresponding logon information.

**To change the size of tempdb in SQL Server 6.5**

# Installing SQL Server

## Estimating the Disk Space Required for Upgrading

Before you perform an upgrade of Microsoft® SQL Server™ version 6.5 to SQL Server 2000, ensure that there is available disk space. This is important if you intend to perform either a one-computer or a two-computer upgrade.

The SQL Server Upgrade Wizard estimates the disk space necessary to upgrade the SQL Server 6.5 server to SQL Server 2000. The wizard examines the current SQL Server 6.5 installation and estimates the amount of disk space the SQL Server 6.5 data will occupy in SQL Server 2000.

You can estimate:

- The size of SQL Server 2000 databases.
- The size of SQL Server 2000 logs.
- The amount of disk space required for **tempdb**.

**Note** The SQL Server Upgrade Wizard estimates the disk space required; it cannot give an exact requirement.

**To estimate the disk space required for an upgrade**

## Installing SQL Server

## Data and Object Transfer

The **Data and Object Transfer** screen allows you to choose upgrade options.

### Export from 6.5 Server / Import

The objects and data check boxes indicate that the SQL Server Upgrade Wizard exports catalog data, objects, and user data from selected Microsoft® SQL Server™ version 6.5 databases and imports them into newly created SQL Server 2000 databases.

### Data Transfer Method

You can perform an upgrade using either of the following data transfer methods:

- **Named pipe** (simultaneous import/export)

A direct pipeline enables the SQL Server Upgrade Wizard to transfer data in memory from Microsoft SQL Server version 6.5. This data transfer method is the most reliable and provides the best performance. However, when performing a one-computer upgrade, you cannot reuse the disk space occupied by the SQL Server 6.5 devices until the version upgrade process is complete, so use this option only if you have disk space available.

- **Tape** (requires a Microsoft Windows NT® tape driver to be installed)

The SQL Server Upgrade Wizard backs up to tape all of the SQL Server 6.5 databases you have selected to upgrade. The SQL Server Upgrade Wizard then optionally deletes all of the SQL Server 6.5 devices, freeing disk space before new data files are created.

**IMPORTANT** The SQL Server Upgrade Wizard deletes all of the SQL Server 6.5 devices, not only those upgraded. You should upgrade all databases if you choose to delete the SQL Server 6.5 devices.

The tape backup option should be used only when you want to upgrade on a single computer but there is not enough space on the hard disk to install SQL Server 2000 alongside SQL Server 6.5 and perform the

version upgrade.

**Note** The SQL Server Upgrade Wizard uses a named pipe, even when performing a tape backup upgrade. SQL Server 6.5 and SQL Server 2000 must be set to listen to the default named pipe, `\\.\pipe\sql\query`.

## Verification

The transfer of objects and data by the SQL Server Upgrade Wizard is a very reliable process. If any objects could not be imported due to errors in those objects or compatibility problems with Microsoft SQL Server, they are noted in the output logs of the SQL Server Upgrade Wizard.

The SQL Server Upgrade Wizard also offers the following optional verification measures:

- **Validate successful object data transfer**

The SQL Server Upgrade Wizard examines the SQL Server 6.5 databases before the upgrade process and SQL Server 2000 databases after the upgrade. For each, the wizard prepares a list of all objects, including schema and stored procedures, and the number of rows in each table. The wizard then compares the two lists and reports any discrepancies.

- **Exhaustive data integrity verification**

The SQL Server Upgrade Wizard performs a checksum for each column of each table before and after the upgrade to verify that data values have not changed.

**Note** The SQL Server Upgrade Wizard does not report as errors any intentional differences in objects. If some objects, typically stored procedures, could not import due to errors in the objects or compatibility problems with SQL Server 2000, they are reported twice: once in the SQL scripts that show the source code of the objects and the error messages received from SQL Server 2000 when trying to create them, and then again in the output of the verification processes.

# Installing SQL Server

## Order of Upgrade Using a Direct Pipeline or Tape Drive

The SQL Server Upgrade Wizard performs a version upgrade using the options specified. The Microsoft® SQL Server™ version 6.5 server and data used by SQL Server 6.5 databases are left intact throughout the version upgrade process. At this time, the SQL Server 6.5 catalog data, objects, and databases are converted so that they are compatible with SQL Server 2000. After the version upgrade is complete, SQL Server 2000 becomes your production system.

The order of upgrade is basically the same for both a direct pipeline and a tape drive upgrade. The one difference is in how data is exported and imported. When using a tape drive, data is exported to the tape drive after shutting down SQL Server 6.5 and before starting SQL Server 2000. This data is then imported from the tape drive later to SQL Server 2000. When using a direct pipeline, the export and import steps are combined in one step, simultaneously.

The following list shows the order in which the SQL Server Upgrade Wizard performs the upgrade from SQL Server 6.5 to SQL Server 2000. The differences between the direct pipeline and tape drive methods are noted.

- Starts SQL Server 6.5
- Updates ODBC and SQL-DMO components on SQL Server 6.5
- Examines SQL Server 6.5 databases
- Exports replication settings
- Exports server configuration settings from the **master** database
- Exports logon information

- Exports database owners
- Exports SQL Executive objects and settings from the **msdb** database
- Exports database objects for all databases chosen
- Shuts down SQL Server 6.5
  - Tape Drive only: Exports data to tape
  - Tape Drive only: Backs up and then deletes SQL Server 6.5 devices
- Starts SQL Server 2000
- Creates databases
- Modifies SQL Executive objects and settings to SQL Server 2000 formats
- Imports logon information
- Imports database objects
  - Tape Drive only: Imports data from tape into SQL Server 2000
  - Direct Pipeline only: Simultaneously exports data from SQL Server 6.5 and imports it into SQL Server 2000
- Imports modified SQL Executive objects and settings into SQL Server 2000
- Imports replication settings

- Examines SQL Server 2000 databases
- Verifies that the upgrade is successful
- Sets database options in SQL Server 2000
- Marks server and databases as moved
- Drops temporary **tempdb** files

## Installing SQL Server

## Upgrading Using One or Two Computers (Logon Screen)

The upgrade process can take place on a single computer or from one computer to another, depending on where Microsoft® SQL Server™ version 6.5 and SQL Server 2000 are installed. The SQL Server Upgrade Wizard identifies the two servers as the export server and import server.

- For a one-computer upgrade, leave the import and export servers at their default values.
- For a two-computer upgrade, select the name of the computer with your SQL Server 6.5 server as the export server. To upgrade SQL Server from one computer to another, the two computers must be in the same network domain.

**IMPORTANT** The one-computer upgrade is the only method supported when upgrading a server used in replication. A two-computer upgrade is not supported for replication servers.

### Export server (6.5)

Export server (6.5) is the name of the SQL Server 6.5 server. This defaults to the name of the computer on which the SQL Server Upgrade Wizard is run, but may be changed if your SQL Server 6.5 server is on another computer.

- **Server name**

Server name is the name of your SQL Server version 6.5 server. This defaults to the name of the computer on which the SQL Server Upgrade Wizard is run, but may be changed if your SQL Server 6.5 server is on another computer.

- **Administrator password ('sa')**

Enter the system administrator (**sa**) password for the SQL Server 6.5 server.

- **Optional startup arguments**

Enter any trace flags or other startup parameters to be used when the SQL Server Upgrade Wizard starts the SQL Server 6.5 server.

## **Import server**

The import server is the name of the SQL Server 2000 server. This is always the name of the computer on which the SQL Server Upgrade Wizard is run.

- **Server name**

Server name is the name of your SQL Server 2000 server computer. This is always the name of the computer on which the SQL Server Upgrade Wizard is run.

- **Administrator password ('sa')**

Enter the system administrator (**sa**) password for the SQL Server 2000 server. Unless you have changed it since installing SQL Server 2000, the default **sa** password is blank.

- **Optional startup arguments**

Enter any trace flags or other startup parameters to be used when the SQL Server Upgrade Wizard starts the SQL Server 2000 server.

# Installing SQL Server

## Selecting a Scripting Code Page

The SQL Server Upgrade Wizard requires the selection of a scripting code page, which is used to create the upgrade scripts. When the **Code Page Selection** screen appears in the Upgrade Wizard, most users can accept the default code page, which is the code page recorded in the **master** database.

In some cases, the actual code page used for a Microsoft® SQL Server™ 6.5 installation differs from the code page recorded in the **master** database. If you know that the actual code page is different from the recorded code page, select the actual code page in the list on the **Code Page Selection** screen.

**CAUTION** If you choose a scripting code page other than the default, do not upgrade replication settings. If the server is involved in replication, reconfigure the replication settings after the upgrade is complete.

The enhancements to collation settings in SQL Server 2000 do not apply directly to this selection of a code page for the SQL Server 6.5 upgrade. For more information about collation enhancements, see [Collations](#).

# Installing SQL Server

## Selecting Databases to Upgrade

When running the SQL Server Upgrade Wizard, you can choose to upgrade some or all Microsoft® SQL Server™ version 6.5 databases. The **master**, **msdb**, and **publication** system databases, as well as the **pubs** and **Northwind** sample databases, are not explicitly available for selection. However, the **master**, **msdb**, and **publication** databases can be selected for upgrading (the default) in the **Server Configuration** dialog box of the SQL Server Upgrade Wizard.

**Note** If you run the SQL Server Upgrade Wizard again after databases have been upgraded, previously updated databases will default to the excluded list. If you want to upgrade a database again, drop the database in SQL Server 2000 and move it to the included list in the wizard.

## Installing SQL Server

## Database Configuration

Before any data is transferred, the SQL Server Upgrade Wizard creates, if necessary, database and log files large enough to contain the upgraded database data. On the **Database Creation** screen there are several options for creating the Microsoft® SQL Server™ 2000 database and log files.

### Using the Default Database Configuration

The SQL Server Upgrade Wizard estimates how much disk space is necessary to hold transferred objects and data for each selected database and creates database files of the estimated sizes. The wizard makes no allowance for free space beyond the loaded data. By default, the data file for a database is placed in the same location as the first device used by that database in SQL Server 6.5.

The SQL Server Upgrade Wizard also creates a log file for each database using the SQL Server 6.5 log size. By default, the log file is placed in the same location as the first device used for log space in SQL Server 6.5.

You can view and edit the default database configuration in the SQL Server Upgrade Wizard. For each database and log file you can modify:

- The name and file path.
- The initial size of the file.
- The autogrow increment.

If using multiple devices in a SQL Server version 6.5 database, then multiple database files are created in the same location. However, the first database file is sized to accommodate the bulk of the data, and the other files are minimally sized. If you want to remove these files, you must do so before they are created. All files are set to grow automatically if extra space is required.

### Using a Custom Database Configuration

You can specify a custom configuration in two ways:

- Using databases and logs that you created in SQL Server 2000.

The SQL Server Upgrade Wizard does not create any user databases. You must create the necessary databases and logs in SQL Server 2000 before you start the SQL Server Upgrade Wizard. Use this option only if necessary.

- Using an SQL script file that you provide.

The SQL Server Upgrade Wizard uses an SQL script file that you provide to create the necessary user databases and logs. Use this option only if you are familiar with the new CREATE DATABASE statement in SQL Server 2000.

If you create the user databases or an SQL script file, the SQL Server 2000 databases must have the same names as in SQL Server 6.5. Also, remember that data may take up more disk space in SQL Server 2000 than in SQL Server 6.5. The SQL Server Upgrade Wizard estimates this growth. You can view the proposed layout of the SQL Server 2000 data files to see the estimated initial size of the SQL Server 2000 database, and edit the default configuration, if necessary. For more information, see [Proposed Database Layout](#).

It is recommended that you leave the autogrow feature on for each database. You may also want to set a backward compatibility level for each database.

### **To edit the default database configuration**

# Installing SQL Server

## Proposed Database Layout

The **Proposed Database Layout** dialog box lists the databases, file groups, and data files that the Microsoft® SQL Server™ Upgrade Wizard will create. You can create or remove file groups and data files from the **File** menu. Double-click a data file to edit the file name, initial size, or file growth details.

### Object Details

Click on a file group or data file in the proposed database layout to view details. Click a database in the proposed database layout to view summary information.

### Drive Summary

The drive summary lists all local fixed-disk drives. For each drive, the existing SQL Server version 6.5 data file size, proposed SQL Server 2000 data file size, and free space are listed. On the **Options** menu, select **Freespace includes 6.5 files** to view the free space that would exist if the SQL Server 6.5 data files were deleted. This option shows the disk space available if the upgrade is performed using tape and the SQL Server 6.5 devices are deleted.

### See Also

[Database Configuration](#)

## Installing SQL Server

## Tape Upgrade Transfer Options

When you perform a tape backup, you must select a tape drive and choose how the SQL Server Upgrade Wizard handles backing up and deleting objects in the Microsoft® SQL Server™ version 6.5 databases.

### Device for Data Transfer

The SQL Server Upgrade Wizard transfers all of the data you are upgrading to this tape drive before the SQL Server 2000 databases are created.

### Backing Up the SQL Server 6.5 Devices

You may also choose to back up the SQL Server 6.5 devices. This is separate from the transfer to tape that the SQL Server Upgrade Wizard uses to complete the upgrade. There are two options for backing up the devices:

- **Prompt me to backup my devices manually**

Before data is exported, the SQL Server Upgrade Wizard pauses and prompts you to perform a backup. The SQL Server Upgrade Wizard does not perform a backup for you. You must use a backup utility such as Microsoft Windows NT® Backup.

- **Automatically copy device files to the following location**

Before data is exported, the SQL Server Upgrade Wizard copies the device files to a shared network directory.

**WARNING** If you back up the devices to tape, remove the tape backup and insert a blank tape before continuing. Before the SQL Server Upgrade Wizard begins transferring data to the tape drive, it formats the tape in the drive. If you do not remove your tape backup, the SQL Server Upgrade Wizard overwrites it.

### Deleting the SQL Server 6.5 Devices

If you decide to delete your SQL Server 6.5 devices before creating the SQL Server 2000 databases, you can choose whether to be prompted before the

devices are deleted. All of the SQL Server 6.5 device files will be deleted if you choose to delete devices, even if you are upgrading only one database. This will render the SQL Server 6.5 server unusable until the files are restored.

**Note** If you choose not to delete the devices, you must have enough disk space for both the SQL Server 6.5 and SQL Server 2000 databases. If sufficient space is available, you should use a Named Pipe upgrade instead of a Tape upgrade.

## **See Also**

[How to perform a SQL Server version 6.5 to SQL Server 2000 upgrade using a tape drive \(SQL Server Upgrade Wizard\)](#)

# Installing SQL Server

## System Configuration

On the **System Configuration** screen, you can set options for system objects to transfer, ANSI Nulls, and quoted identifiers.

### System Objects to Transfer

When the SQL Server Upgrade Wizard upgrades the **master** database, it can upgrade several configuration options:

- Server configuration

Logon information and remote logon registrations and server configuration options relevant to Microsoft® SQL Server™ 2000 are transferred as part of the version upgrade process. The SQL Server 6.5 configuration options not used in SQL Server 2000 are not transferred.

- Replication settings

All articles, subscriptions, and publications of each selected database, including the distribution database, if any, are transferred and upgraded.

- SQL Executive settings

All tasks scheduled by SQL Executive are transferred and upgraded so that the SQL Server 2000 can schedule and run the tasks in SQL Server Agent.

**Note** Upgrading replication or SQL Executive settings causes existing modifications made to the SQL Server 2000 replication or SQL Server Agent settings to be overwritten.

### ANSI Nulls

The ANSI\_NULLS option controls both database default nullability and comparisons against null values. When upgrading Microsoft SQL Server version 6.5 to the SQL Server 2000, set the ANSI\_NULLS option to ON or OFF.

When the SQL Server Upgrade Wizard creates the SQL Server 2000 database tables, the database default nullability determined by the ANSI\_NULLS option

is not an issue. All columns are explicitly qualified as NULL or NOT NULL based on their status in SQL Server 6.5.

The ANSI\_NULLS option is important with regard to comparisons against null values, when the SQL Server Upgrade Wizard creates the SQL Server 2000 database objects. With ANSI\_NULLS set to ON, the comparison operators EQUAL (=) and NOT EQUAL (<>) always return NULL when one of its arguments is NULL. With ANSI\_NULLS set to OFF, these operators return TRUE or FALSE, depending on whether both arguments are NULL.

In SQL Server 6.5, the ANSI\_NULLS option in objects, such as stored procedures and triggers, is resolved during query execution time. In SQL Server 2000, the ANSI\_NULLS option is resolved when the object is created. You must choose the ANSI\_NULLS option setting you want for all objects in the databases you are upgrading. The SQL Server Upgrade Wizard then creates all database objects using this ANSI\_NULLS setting.

## Quoted Identifiers

**Note** Quoted identifiers are used by default in SQL Server 2000, that is, they are set to ON. This is different from SQL Server 7.0 where they were set to OFF by default.

The QUOTED\_IDENTIFIER setting determines what meaning Microsoft SQL Server gives to double quotation marks ("). When QUOTED\_IDENTIFIER is set to OFF, double quotation marks delimit a character string, just as single quotation marks do. When QUOTED\_IDENTIFIER is set to ON, double quotation marks delimit an identifier, such as a column name. An identifier must be enclosed in double quotation marks; for example, if its name contains characters that are otherwise not allowed in an identifier, including spaces and punctuation, or if the name conflicts with a reserved word in Transact-SQL. Regardless of the QUOTED\_IDENTIFIER setting, an identifier can also be delimited by square brackets.

The meaning of the following statement, for example, depends on whether QUOTED\_IDENTIFIER is set to ON or OFF:

```
SELECT "x" FROM T
```

If QUOTED\_IDENTIFIER is set to ON, "x" is interpreted to mean the column

named **x**. If it is set to OFF, "x" is the constant string x and is equivalent to the letter x.

If the previous SELECT statement example were part of a stored procedure created when QUOTED\_IDENTIFIER was set to ON, then "x" would always mean the column named **x**. Even if the QUOTED\_IDENTIFIER setting was later switched, and set to OFF, the stored procedure would respond as if it were set to ON and treat "x" as the column named **x**.

When the SQL Server Upgrade Wizard re-creates database objects in SQL Server 2000, the QUOTED\_IDENTIFIER setting determines how all of these objects behave. If all database objects were created in SQL Server 6.5 with the same QUOTED\_IDENTIFIER setting, click that setting, either **On** or **Off**. If objects were created in SQL Server version 6.5 with a mix of the two settings, or if you are unsure of the settings used, click **Mixed**.

With the **Mixed** option, the SQL Server Upgrade Wizard first converts all objects containing double quotation marks with QUOTED\_IDENTIFIER set ON. The SQL Server Upgrade Wizard then converts any objects that failed to be created with QUOTED\_IDENTIFIER set OFF.

## **See Also**

[How to perform a SQL Server version 6.5 to SQL Server 2000 upgrade using a direct pipeline \(SQL Server Upgrade Wizard\)](#)

[How to perform a SQL Server version 6.5 to SQL Server 2000 upgrade using a tape drive \(SQL Server Upgrade Wizard\)](#)

# Installing SQL Server

## Completing the SQL Server Upgrade Wizard

Use this screen to view the summary of choices you have made.

Click **View warnings and choices in notepad** to open a text version of the upgrade script. If all options are correct, click **Finish**.

# Installing SQL Server

# Upgrade Script Interpreter

After you click **Finish**, this screen displays the progress of the upgrade.

## Progress indicator

Displays information about the current task and its progress toward completion. The information presented varies according to the type of task.

## Task

The SQL Server Upgrade Wizard adds each upgrade task to the list as it is started.

## Status

The SQL Server Upgrade Wizard displays the status (Running, Done, or Error) for each task.

## Started

The SQL Server Upgrade Wizard displays the time and date on which the task began.

## End

The SQL Server Upgrade Wizard displays the time and date on which a completed or terminated task is finished.

## Pause Task

Temporarily suspends the version upgrade process until you click **Resume**.

## Cancel Task

Cancels the currently running task and proceeds to the next task. Do not cancel a task unless you are certain the current task does not need to be completed before subsequent tasks are run.

## Retry Task

Retries the current upgrade task. If a task ended in an error and you corrected the problem, the SQL Server Upgrade Wizard retries the current task.

## **Pause Between Steps**

Allows you to participate interactively in the version upgrade process and track the progress of the SQL Server Upgrade Wizard. The SQL Server Upgrade Wizard asks for confirmation between each step of the version upgrade process.

# Installing SQL Server

# Backward Compatibility

Backward compatibility issues are divided in these sections:

- For issues related to upgrades from Microsoft® SQL Server™ version 7.0 to SQL Server 2000, see:

[SQL Server 2000 and SQL Server version 7.0](#)

- For issues related to upgrades between SQL Server 6.5 and Microsoft SQL Server 2000, see:

[SQL Server 2000 and SQL Server version 6.5](#)

If upgrading from SQL Server 6.5 to SQL Server 2000, review both sections.

# Installing SQL Server

## SQL Server 2000 and SQL Server version 7.0

Microsoft® SQL Server™ 2000 is compatible with SQL Server 7.0 in most ways. The section describes backward compatibility issues when upgrading from SQL Server 7.0 to SQL Server 2000:

- Client Network Utility and Named Instances
- Multiserver Jobs and Named Instances
- Upgrading SQL Server 6.5 Client Software
- Authentication Modes
- ROWCOUNT Setting for Operations Against Remote Tables
- Server Configuration Options
- Recovery Models and Database Options
- Reserved Keywords
- SQL Profiler Extended Stored Procedures
- Default Connection Option Settings in SQL Query Analyzer
- **bcp** Utility
- Database Diagrams from Earlier Versions of Visual Database Design Tools

- Data Transformation Services
- Specifying Trusted Connections
- Extended Objects in SQL-DMO
- SQL-SCM
- English Query and SQL Server 7.0 OLAP Services

## **Client Network Utility and Named Instances**

When using the SQL Server client connectivity components from SQL Server 7.0 or earlier, you must set up an alias using the Client Network Utility before you connect to a named instance of SQL Server 2000. For example, on a SQL Server 7.0 client, to connect to a named instance of SQL Server 2000, you must add an alias that points to

`\\computername\pipe\MSSQL$instancename\sql\query`. If you use an alias name of `computername\instancename`, clients can connect by specifying this name in the same way as SQL Server 2000 clients do. For the TCP/IP Sockets and NWLink IPX/SPX Net-Libraries, you must use the Client Network Utility to define an alias on the client that specifies the port address on which the named instance is listening.

## **Multiserver Jobs and Named Instances**

When using Master Servers and Target Servers, SQL Server 7.0 cannot interoperate with named instances of SQL Server 2000. To use an instance of SQL Server 7.0 with an instance of SQL Server 2000 for MSX/TSX operations, you must use a default instance, not a named instance, of SQL Server 2000.

## **Upgrading SQL Server 6.5 Client Software**

When running an instance of SQL Server version 6.5 on a server, this issue

applies:

If you are upgrading from SQL Server 6.5 client software to SQL Server 2000 client software (and you have an application that uses the default Net-Library), you must use the Client Network Utility to make either Named Pipes or Multiprotocol the default Net-Library to make Windows Authentication connections.

## **Authentication Modes**

SQL Server 2000 can operate in one of two security (authentication) modes:

- Windows Authentication Mode (Windows Authentication)
- Mixed Mode (Windows Authentication and SQL Server Authentication)

Mixed Mode allows users to connect using Windows Authentication or SQL Server Authentication. Users who connect through a Microsoft Windows NT® 4.0 or Windows 2000 user account can make use of trusted connections (connections validated by Windows NT 4.0 or Windows 2000) in either Windows Authentication Mode or Mixed Mode.

SQL Server Authentication is provided for backward compatibility. An example of SQL Server Authentication would be if you create a single Microsoft Windows® 2000 group, add all necessary users to that group, and then grant the Windows 2000 group login rights to SQL Server and access to any necessary databases.

## **ROWCOUNT Setting for Operations Against Remote Tables**

ROWCOUNT is not supported for INSERT statements against remote tables in SQL Server 2000 when the database compatibility level is set to 80. For these INSERT operations, the SET ROWCOUNT option is ignored.

The ROWCOUNT setting for INSERT statements against remote tables was supported in SQL Server 7.0.

## **Server Configuration Options**

These server configuration options are not supported in SQL Server 2000.

<b>default sortorder id</b>	<b>resource timeout</b>
<b>extended memory size</b>	<b>spin counter</b>
<b>language in cache</b>	<b>time slice</b>
<b>language neutral full-text</b>	<b>unicode comparison style</b>
<b>max async IO</b>	<b>unicode locale id</b>

For more information about configuration options, see [Setting Configuration Options](#) and [sp\\_configure](#).

## Recovery Models and Database Options

Microsoft® SQL Server™ 2000 provides the following recovery models to simplify recovery planning, simplify backup and recovery procedures, and to clarify tradeoffs between system operational requirements:

- Simple Recovery
- Full Recovery
- Bulk-Logged Recovery

Each model addresses different needs for performance, disk and tape space, and protection against data loss.

In SQL Server 7.0 and earlier, similar functionality was provided through the combined settings of the **trunc. log on chkpt** and **select into/bulkcopy** database options, which could be set using the **sp\_dboption** stored procedure.

This table maps the settings of **trunc. log on chkpt** and **select into/bulkcopy** to the new recovery models.

<b>If trunc. log on chkpt is:</b>	<b>And select into/bulkcopy is:</b>	<b>The recovery model is:</b>
FALSE	FALSE	FULL

FALSE	TRUE	BULK-LOGGED
TRUE	TRUE	SIMPLE
TRUE	FALSE	SIMPLE

**Note** If you upgrade a database in which the **trunc. log on chkpt** and **select into/bulkcopy** options are set to TRUE, **select into/bulkcopy** is set to FALSE, forcing the database into the simple recovery model.

The **trunc. log on chkpt** and **select into/bulkcopy** database options are supported in SQL Server 2000 for backward compatibility purposes, but may not be supported in future releases.

In SQL Server 2000, the ALTER DATABASE Transact-SQL statement provides a SET clause for specifying database options, including recovery models. For more information about database options, see [Setting Database Options](#) and [ALTER DATABASE](#).

## Reserved Keywords

These words are no longer reserved keywords in SQL Server 2000: AVG, COMMITTED, CONFIRM, CONTROLROW, COUNT, ERROREXIT, FLOPPY, ISOLATION, LEVEL, MAX, MIN, MIRROREXIT, ONCE, ONLY, PERM, PERMANENT, PIPE, PREPARE, PRIVILEGES, REPEATABLE, SERIALIZABLE, SUM, TAPE, TEMP, TEMPORARY, UNCOMMITTED, WORK.

These words are reserved keywords in SQL Server 2000: COLLATE, FUNCTION, OPENXML.

## SQL Profiler Extended Stored Procedures

SQL Profiler extended stored procedures, such as **xp\_trace\_addnewqueue** and **xp\_trace\_generate\_event**, are not supported in SQL Server 2000. They have been replaced by a set of new stored procedures and system user-defined functions. For more information, see [Creating and Managing Traces and Templates](#).

## Default Connection Option Settings in SQL Query Analyzer

In SQL Server version 7.0 and earlier, the default setting for SET QUOTED\_IDENTIFIER in SQL Query Analyzer was OFF. In SQL Server 2000, the default setting in SQL Query Analyzer is ON, which is also the default setting for ODBC and OLE DB. Moreover, several new features in SQL Server 2000, such as indexed views and indexes on computed columns, require this option to be ON.

**Note** If you use double quotation marks for strings when QUOTED\_IDENTIFIER is ON, you will receive a syntax error.

## **bcp Utility**

To read character files created by earlier versions of DB-Library **bcp** in SQL Server 2000, use the **-V** switch. For more information, see [bcp Utility](#).

## **Database Diagrams from Earlier Versions of Visual Database Design Tools**

For users who have database diagrams created with earlier versions of the visual database design tools:

- If the first visual database tool that was used against a SQL Server 2000 database is a version earlier than the tools in SQL Server 2000, SQL Server Enterprise Manager will not be able to open or create a database diagram in that database. Any attempt to do so results in the error: ODBC error: [Microsoft][ODBC SQL Server Driver][SQL Ser

There are several visual database tools that can put a database into this state. These include the Query Designer, the View Designer, the Database Designer, and the Table Designer in SQL Server 7.0 and earlier, as well as many tools that enumerate the objects in a database. These tools are also in Microsoft Access 2000 and Microsoft Visual Studio® 6.

Running the following script on the database allows SQL Server Enterprise Manager to work with the database diagrams in that database:

```
alter table dbo.dproperties add uvalue nvarchar(255) null
```

```
go
if exists(select * from dbo.dtproperties) exec('update dbo.dtpro]
go
```

After this script has been run, both the SQL Server Enterprise Manager in SQL Server 2000 and the earlier versions of the visual database tools can jointly access the database diagrams in the database. There are additional issues to consider when using the earlier versions of the database tools against a SQL Server 2000 database. For more information, see [Hardware and Software Requirements for Installing SQL Server 2000](#).

## Data Transformation Services

These are the backward compatibility issues for Data Transformation Services (DTS).

### Extended DTS Objects

Some objects in Data Transformation Services (DTS) are extended in SQL Server 2000. For more information about using new Data Transformation Services objects, methods, and properties with SQL Server 7.0 and earlier, see [Extended DTS Objects](#).

### Copy SQL Server Objects Task

There are restrictions on using the Copy SQL Server Objects task (Transfer SQL Server Objects task in SQL Server version 7.0) when copying database objects between an instance of SQL Server 2000 and SQL Server 7.0. For more information, see [Copy SQL Server Objects Task](#).

### Running DTS Packages on SQL Server 7.0 or Earlier

DTS packages created on an instance of SQL Server 2000 cannot be loaded or run on an instance of SQL Server version 7.0 or earlier. If you attempt to do this, you may receive one of the following messages:

- "Invalid class string."

- "Parameter is incorrect."

Both messages indicate that the current server does not contain all the components necessary to load the package and cannot support objects defined in the DTS package, such as tasks and transformations.

However, if you receive one of these messages, you can still open and run the package on an instance of SQL Server 2000.

## **Using DTS with Different Collations, Different Code Pages, and Non-Unicode Data**

When using the Copy SQL Server Objects task and Copy Column transformation to copy non-Unicode data between an instance of SQL Server 2000 and SQL Server 7.0, issues arise when using different code pages and collations. For more information, see [Data Conversion and Transformation Considerations](#).

## **Specifying Trusted Connections**

In SQL Server 7.0, you did not have to code "trusted\_connection=yes" in your connection strings for ADO, OLE DB, or ODBC to obtain a trusted connection. If you did not specify a UID and PASSWORD, SQL Server would default to trying a trusted connection. In SQL Server 2000, you must code "trusted\_connection=yes" to obtain trusted connection.

## **Extended Objects in SQL-DMO**

Some objects in SQL-DMO are extended in SQL Server 2000. For more information about using extended SQL-DMO objects, methods, and properties with SQL Server 7.0 or earlier, see [Programming Extended SQL-DMO Objects](#).

## **SQL-SCM**

The SQL-SCM (Service Control Manager) API has been removed and is no longer supported.

## **English Query and OLAP Services for SQL Server 7.0**

For users of OLAP Services for SQL Server 7.0 who want to install or uninstall English Query, these issues apply:

- OLAP Services for SQL Server 7.0 must not be running during installation. Shut down the OLAP Services service before installing English Query. (See the Services application in Control Panel.)
- If you have installed OLAP Services for SQL Server 7.0 and you uninstall English Query, you must reinstall OLAP Services. Conversely, if you have installed English Query and you uninstall OLAP Services, you must reinstall English Query to maintain OLAP connectivity.

These issues do not occur with SQL Server 2000 Analysis Services (formerly OLAP Services).

# Installing SQL Server

## SQL Server 2000 and SQL Server version 6.5

Microsoft® SQL Server™ 2000 is compatible with SQL Server version 6.5 in many respects. Most product functionality of SQL Server version 6.5 remains in SQL Server 2000. Most applications for SQL Server 6.5 work unchanged after the database server is upgraded to SQL Server 2000 by the SQL Server Upgrade Wizard.

The SQL Server 2000 upgrade process:

- Adds functionality, either new to SQL Server 2000 or changed from earlier versions, which makes tasks easier to accomplish.
- Minimizes the time and effort needed to upgrade.

In some cases, compatibility issues can arise:

- Configuration Options

Some server configuration options have changed.

- SQL-DMO, Tasks, and Replication

Task, replication, and device objects have changed. SQL Server 2000 uses jobs instead of tasks, and provides new system tables and system stored procedures.

- Replication and Triggers

Replication types that allow data modifications at the Subscriber use triggers to track changes to published tables. If there are triggers on your application that modify published tables, the **sp\_configure** server option **nested triggers** should be enabled. This option affects tables used in merge replication or tables used in snapshot replication or transactional replication with the immediate updating or queued updating option. Before adding these types of replication to an existing database that uses triggers, be sure your application works correctly with the **nested triggers** option enabled. The **nested triggers** option is enabled by default; however, if this option was disabled previously, you will need to enable it again.

- Segments and Devices

SQL Server 7.0 and SQL Server 2000 use files and filegroups instead of segments and devices for storing indexes or tables. Unless your application depends upon the physical layout of segments within devices, this does not create compatibility problems for your application.

- System Tables

If your applications depend upon accessing system tables directly, the applications may need to be revised. It is recommended that you use system stored procedures or information schema views.

Here are the SQL Server 6.x system tables that are not included with SQL Server 2000.

<b>master.dbo.spt_datatype_info</b>	<b>sysprocedures</b>
<b>sysbackupdetail</b>	<b>sysrestoredetail</b>
<b>sysbackuphistory</b>	<b>sysrestorehistory</b>
<b>syshistory</b>	<b>syssegments</b>
<b>syskeys</b>	<b>systasks</b>
<b>syslocks</b>	<b>sysusages</b>

- Backup and Restore

SQL Server 2000 uses BACKUP and RESTORE statements in place of DUMP and LOAD. DUMP and LOAD are supported for backward compatibility, but with some limitations.

- System Stored Procedures

Some system stored procedures are no longer supported.

For more information, see the discussion of specific backward compatibility issues.

## **Setting a Backward Compatibility Level**

When running at its default settings, Microsoft SQL Server 2000 implements SQL-92 behaviors for some Transact-SQL statements whose behaviors differed from the standard in earlier versions of SQL Server. SQL Server 2000 also enforces reserved keywords that were not keywords in earlier versions of SQL Server. If upgrading existing systems with existing applications, you can use the database compatibility level settings to retain the earlier behaviors if your existing applications depend on those behaviors. This gives you time to upgrade applications in an orderly fashion. Most applications, however, are not affected by the changes in behavior and work at the SQL Server 2000 compatibility level.

The compatibility level is specified for each database using the **sp\_dbcmptlevel** system stored procedure. The database compatibility level can be set to 60 (version 6.0 compatibility), 65 (version 6.5 compatibility), 70 (version 7.0 compatibility), and the default 80 (SQL Server 2000 compatibility). The effects of the compatibility level settings are generally limited to the behaviors of a small number of Transact-SQL statements that also existed in earlier versions of SQL Server. Even when the database compatibility level is set to 60 or 65, applications gain almost all of the benefits of the new performance enhancements of SQL Server 2000. Applications still benefit from features such as the improved query processor. For more information, see the discussion of specific behaviors controlled by the different settings in [sp\\_dbcmptlevel](#).

For installations of all instances of SQL Server 2000, the default level for all databases is 80. For upgrades from SQL Server 7.0 to SQL Server 2000, the default level for all databases is 80. For upgrades from SQL Server 6.5 and SQL Server 6.0 to SQL Server 2000, the existing default compatibility level is retained.

**IMPORTANT** The compatibility level for the **master** database is 80 and cannot be changed. If you have added any user-defined objects to **master**, you must ensure they work correctly at the 80 compatibility level.

The **model** database is set automatically to the SQL Server 2000 compatibility level during an upgrade. All new user-defined databases are created with the same compatibility level setting as **model**. If you do not want to use any SQL Server 2000 behavior in new databases created after an upgrade, use **sp\_dbcmptlevel** to change the compatibility level setting in **model**.

Certain behaviors are not enabled at lower compatibility levels. For example, the keywords LEFT, OUTER, and JOIN are not keywords at compatibility level 60.

This means the database compatibility level must be set to 65 or higher before the LEFT OUTER JOIN clause becomes valid. Before any applications can take advantage of features only available at a higher compatibility level, all applications using the database must be upgraded to work correctly at the higher compatibility level.

Likewise, setting the compatibility level of a database to 65 makes the database version-6.5 compatible, but does not necessarily provide version 6.5 behaviors. For example, when SET ANSI\_PADDING is ON and you attempt to insert the strings 'abc' and 'abc ' into a primary key column, SQL Server 2000 considers the strings to be duplicates and does not violate the primary key constraint. In SQL Server 6.5, the two strings are considered to be unique and both insertions succeed. Setting the compatibility level to 65 does not force SQL Server 2000 to treat the strings as unique values.

**Note** While running at compatibility level 60 or 65 preserves legacy behaviors on SQL Server 2000, support for these behaviors may be dropped in future versions of SQL Server. It is recommended that you plan to upgrade your applications to work correctly with the compatibility level set to 80 as soon as is practicable.

## **See Also**

[Reserved Keywords](#)

[System Stored Procedures](#)

[System Tables](#)

## SQL Server Backward Compatibility Details

Microsoft® SQL Server™ 2000 adds many new features. Most of the changes are internal and will not affect your database scripts or applications. All Transact-SQL statements are compatible. However, administration tools or scripts should be updated to work with SQL Server 2000.

The backward compatibility topics in this section contain a detailed list of features and behaviors supported in SQL Server version 6.5 that have changed and could possibly affect your administration tools or scripts. These changes are not controlled by the backward compatibility level.

To indicate their potential effect on administration tools or scripts, feature changes have been grouped into four levels.

Level	Consists of
1	Administrative statements, stored procedures, or SQL Server items that have been removed from, or are no longer supported in, SQL Server 2000. Administrative tools or scripts using these items must be fixed prior to using SQL Server 2000. For more information about these features, see <a href="#">Level 1: Handling Discontinued Functionality</a> .
2	Important changes that produce different behavior from earlier versions of SQL Server. For example, items in this category are those that have changed behavior in data type conversion or usage of selected functions, changed behavior of clauses in selected Transact-SQL statements and stored procedures, changed column names in selected system tables, and changed behavior due to the database compatibility setting. For more information about these features, see <a href="#">Level 2: Handling Major Changes to Behavior</a> .
3	Items supported for backward compatibility only. Any item included in this category is fully supported, but may be removed or unsupported in a future release. SQL Server 2000 provides features that accomplish these tasks more efficiently and have ongoing support. For more information about these features, see <a href="#">Level 3:</a>

	<a href="#">Updating to Improve Earlier Functionality.</a>
4	Minor changes that produce different behavior from earlier versions of SQL Server. For example, items in this category are either ignored or have one or more ignored parameters, changed byte lengths, added parameters or columns, or changed data type columns. For more information about these features, see <a href="#">Level 4: Handling Minor Changes to Behavior</a> .

**Note** You might find it helpful to review [SQL Server 2000 and SQL Server version 7.0](#) as well.

## **SetHostName property not used in SQL Server 2000**

When using SQL Server 6.5 integrated security, SQL Server 6.5 did not report the Windows NT account used by a connection unless the system administrator activated the SET HOSTNAME TO USERNAME option in SQL Enterprise Manager.

The setting could also be activated through the SQL-DMO SetHostName property. With this setting in effect, these functions and columns returned the user's Windows NT account name instead of the network name of the client computer:

- Transact-SQL HOST\_NAME() function
- **hostname** column in the result set returned by **sp\_who**
- **hostname** column in **sysprocesses**

In SQL Server 2000, the **loginame** column in the **sp\_who** result set contains the Windows NT account name for connections made using Windows NT Authentication. Applications needing the Windows NT account associated with a connection using Windows NT Authentication should reference this column.

SQL Server 2000 Enterprise Manager no longer presents the SET HOSTNAME TO USERNAME option. SQL Server 2000 ignores the setting of the SQL-DMO SetHostName property.

## Level 1: Handling Discontinued Functionality

Backward Compatibility Level 1 consists of administrative statements, stored procedures, or Microsoft® SQL Server™ items that were supported in SQL Server 6.5 but have been removed from, or are no longer supported in, SQL Server 2000. Administrative tools or scripts using these items must be fixed prior to using SQL Server 2000.

<b>This subheading</b>	<b>Relates to</b>
<a href="#">Backup and Restore</a>	BACKUP RESTORE DUMP LOAD <b>sysbackuphistory</b> <b>sysbackupdetail</b> <b>sysrestorehistory</b> <b>sysrestoredetail</b> <b>backupfile</b> <b>backupmediafamily</b> <b>backupmediaset</b> <b>backupset</b> <b>restorefile</b> <b>restorefilegroup</b> <b>restorehistory</b>
<a href="#">Configuration Options</a>	<b>sp_configure</b> (backup buffer size, backup threads, database size, free buffers, hash buckets, LE threshold maximum, LE threshold minimum, LE threshold percent, logwrite sleep, max lazywrite IO, memory, open databases, procedure cache, RA cache hit limit, RA cache miss limit, RA delay, RA pre-fetches, RA slots per thread, RA worker threads, recovery flags, remote conn timeout, SMP)

	<b>concurrency, sort pages, min memory per query, index create memory, tempdb in ram, and user connections options) trace flag 204</b>
<a href="#">Custom Sort Orders</a>	Character sets, sort orders, and Unicode collations
<a href="#">Databases</a>	ALTER DATABASE
<a href="#">Database Options</a>	<b>sp_dboption (subscribe and no chkpt. on recovery options) sp_addsubscription RESTORE</b>
<a href="#">Data Access Objects (DAO)</a>	<b>odbcadm</b> utility
<a href="#">DBCC</a>	DBCC DBREINDEX DBCC MEMUSAGE DBCC SHRINKDB
<a href="#">DB-Library</a>	Two-Phase Commit DB-Library for Visual Basic
<a href="#">DECnet Network Library</a>	DECnet Sockets Net-Library
<a href="#">Disk Commands</a>	DISK REINIT DISK REFIT ALTER DATABASE
<a href="#">Disk Mirroring</a>	DISK MIRROR DISK REMIRROR DISK UNMIRROR
<a href="#">Indexes</a>	CREATE INDEX
<a href="#">Open Data Services</a>	Windows NT Component Services SRV_CONFIG SRV_PROC SRV_SERVER srv.h Opens60.lib
<a href="#">Program Group Tools and Utilities</a>	Client Network Utility ISQL_w MS Query SQL Client Configuration

	SQL Enterprise Manager SQL Help SQL Security Manager SQL Trace SQL Performance Monitor SQL Service Manager SQL Setup SQL Query Analyzer SQL Server Enterprise Manager SQL Server Profiler SQL Server Service Manager
<a href="#">Replication</a>	Restricted publications <b>DBOption</b> object <b>ReplicationDatabase</b> object EnablePublishing property <b>repl_publisher</b> login
<a href="#">Security</a>	DENY Delimited Identifiers
<a href="#">Segments</a>	CREATE INDEX CREATE TABLE <b>sp_addsegment</b> <b>sp_dropsegment</b> <b>sp_extendsegment</b> <b>sp_helpsegment</b> CREATE DATABASE ALTER DATABASE
<a href="#">Services</a>	SQL Executive
<a href="#">SET DISABLE_DEF_CNST_CHK</a>	SET DISABLE_DEF_CNST_CHK
<a href="#">SET SHOWPLAN</a>	SET SHOWPLAN SET SHOWPLAN_ALL SET SHOWPLAN_TEXT
<a href="#">SQL Alerter</a>	SQLALRTR.exe
<a href="#">SQL-DMO</a>	sqlole.dll
<a href="#">System Stored Procedures (General Extended</a>	<b>xp_snmp_getstate</b> <b>xp_snmp_raisetrap</b>

Procedures)	
<a href="#">System Stored Procedures (Replication)</a>	<b>sp_replica</b> <b>sp_replsync</b> <b>sp_helppublicationsync</b> <b>sp_subscribe</b> <b>sp_unsubscribe</b> <b>@@ERROR</b> <b>sp_changepublication</b> <b>sp_addpublisher</b> <b>sp_adddistpublisher</b> <b>sp_droppublisher</b> <b>sp_dropdistpublisher</b> <b>sp_distcounters</b> <b>sp_helpreplicationdb</b> <b>sp_helpreplicationdboption</b> <b>sp_replstatus</b>
<a href="#">System Stored Procedures (System)</a>	ALTER TABLE CREATE TABLE <b>sp_help</b> <b>sp_helpconstraint</b> <b>sp_commonkey</b> <b>sp_dropkey</b> <b>sp_foreignkey</b> <b>sp_helpjoins</b> <b>sp_helpkey</b> <b>sp_primarykey</b> <b>sp_placeobject</b> <b>sp_dbinstall</b> <b>sp_attach_db</b> <b>sp_makestartup</b> <b>sp_unmakestartup</b> <b>sp_procoption</b> <b>sp_helplogins</b> <b>sp_helprotect</b> <b>sp_tableoption</b> <b>sp_serveroption (fallback option)</b> <b>sp_setlangalias</b>

	<b>sp_droplanguage</b> <b>sp_fallback_activate_svr_db</b> <b>sp_fallback_deactivate_svr_db</b> <b>sp_fallback_enroll_svr_db</b> <b>sp_fallback_help</b> <b>sp_fallback_permanent_svr</b> <b>sp_fallback_upd_dev_drive</b> <b>sp_fallback_withdraw_svr_db</b> <b>sp_devoption</b> <b>sp_diskdefault</b> <b>sp_helplog</b> <b>sp_helpstartup</b> <b>sp_help_revdatabase</b> <b>sp_sqlexec</b> <b>sp_addlanguage</b>
<a href="#">System Stored Procedures (Tasks)</a>	<b>sp_addalert</b> <b>sp_addnotification</b> <b>sp_addoperator</b> <b>sp_dropalert</b> <b>sp_dropnotification</b> <b>sp_dropoperator</b> <b>sp_helpalert</b> <b>sp_helphistory</b> <b>sp_helpnotification</b> <b>sp_helpoperator</b> <b>sp_purgehistory</b> <b>sp_runtask</b> <b>sp_stoptask</b> <b>sp_updatealert</b> <b>sp_updatenotification</b> <b>sp_updateoperator</b> <b>sp_add_alert</b> <b>sp_add_notification</b> <b>sp_add_operator</b> <b>sp_delete_alert</b> <b>sp_delete_notification</b>

**sp\_delete\_operator**  
**sp\_help\_alert**  
**sp\_help\_jobhistory**  
**sp\_help\_notification**  
**sp\_help\_operator**  
**sp\_purge\_jobhistory**  
**sp\_start\_job**  
**sp\_stop\_job**  
**sp\_update\_alert**  
**sp\_update\_notification**  
**sp\_update\_operator**

[System Tables](#)

Information Schema Views  
System Stored Procedures (Catalog Procedures)  
**sysdevices** (**mirrorname** and **stripeset** columns)  
**syshistory**  
**sysjobhistory**  
**sysindexes** (**distribution**, **segment**, **rowpage**, **keys1**, and **keys2** columns)  
**syskeys**  
**syslocks**  
**syslockinfo**  
**syslogs**  
**sysprocesses** (**gid** and **suid** columns)  
**sysprocedures**  
**syscomments**  
**syssegments**  
CREATE DATABASE  
ALTER DATABASE  
CREATE TABLE  
ALTER TABLE  
CREATE INDEX  
**systasks**  
**sysjobs**  
**sysjobsteps**  
**sysjobservers**

	<b>sysusages</b> <b>master.dbo.spt_datatype_info</b>
<a href="#">Transactions</a>	Data type conversions
<a href="#">Utilities</a>	<b>probe</b> login

## Level 2: Handling Major Changes to Behavior

Backward Compatibility Level 2 consists of important changes in Microsoft® SQL Server™ 2000 that produce different behavior from earlier versions of SQL Server. For example, items in this category are those that have changed behavior in data type conversion or usage of selected functions, changed behavior of clauses in selected Transact-SQL statements and stored procedures, changed column names in selected system tables, and changed behavior due to the database compatibility setting. This topic covers backward compatibility details for these items.

<b>This subheading</b>	<b>Relates to</b>
<a href="#">Backup and Restore</a>	BACKUP CREATE DATABASE ALTER DATABASE RESTORE <b>sp_dboption</b>
<a href="#">Bulk Copy</a>	<b>bcp</b> Utility
<a href="#">Configuration Options</a>	Setting Configuration Options <b>sp_configure (open objects and user connections options)</b>
<a href="#">Database Pages and Extents</a>	Pages and Extents
<a href="#">Data Types</a>	CAST and CONVERT Data Types
<a href="#">DB-Library</a>	<b>dbcursorfetchex</b>
<a href="#">Empty Strings</a>	<b>sp_dbcmtlevel</b> CHARINDEX DATALENGTH LEFT LTRIM PATINDEX REPLICATE RIGHT RTRIM

	SPACE SUBSTRING UPDATETEXT
<a href="#">Indexes</a>	CREATE INDEX
<a href="#">INSERT</a>	<b>sp_dbcmptlevel</b> INSERT
<a href="#">Keyset Cursors</a>	Keyset cursors
<a href="#">LTRIM and RTRIM Trimming Functions</a>	LTRIM RTRIM
<a href="#">ODBC</a>	<b>SQLGetDiagRec</b> <b>SQLMoreResults</b>
<a href="#">RIGHT</a>	Using Identifiers Reserved Keywords
<a href="#">Security</a>	GRANT REVOKE DENY <b>sp_addlinkedsrvlogin</b>
<a href="#">SELECT</a>	SELECT
<a href="#">SET SHOWPLAN</a>	SET SHOWPLAN_ALL SET SHOWPLAN_TEXT
<a href="#">System Tables</a>	Information Schema Views System Stored Procedures (Catalog Procedures) <b>sysdatabases</b> ( <b>logptr</b> and <b>dumptrdate</b> columns) <b>sysmessages</b> ( <b>langid</b> column) <b>syslogins</b> ( <b>language</b> column) computed columns
<a href="#">Table Hints</a>	DELETE FROM INSERT SELECT UPDATE
<a href="#">Transactions</a>	SET TRANSACTION ISOLATION LEVEL SET CURSOR_CLOSE_ON_COMMIT

	ROLLBACK DECLARE CURSOR
<a href="#">Triggers and System Stored Procedures</a>	<b>sp_dbcmptlevel</b> <b>sp_create_removable</b> CREATE TRIGGER SET QUOTED_IDENTIFIER SET ANSI_NULLS SET ANSI_DEFAULTS
<a href="#">UPDATE</a>	@@ERROR UPDATE INSERT
<a href="#">UPDATETEXT</a>	UPDATETEXT WRITETEXT
<a href="#">Views</a>	DELETE INSERT UPDATE

## Level 3: Updating to Improve Earlier Functionality

Backward Compatibility Level 3 consists of items that were supported in SQL Server version 6.5 but are supported in SQL Server 2000 (and SQL Server 7.0) for backward compatibility only. Any item included in this category is fully supported, but may be removed or unsupported in a future release. It is recommended that, as time allows, the backward compatible item be replaced with the recommended item. SQL Server 2000 provides features that accomplish these tasks more efficiently and have ongoing support.

This topic covers backward compatibility details for these items.

<b>This subheading</b>	<b>Relates to</b>
<a href="#">Backup and Restore</a>	BACKUP RESTORE CREATE DATABASE
<a href="#">Database Options</a>	<b>sp_dboption (publish option)</b> <b>sp_replicationdboption</b>
<a href="#">DBCC</a>	DBCC NEWALLOC DBCC CHECKALLOC DBCC ROWLOCK Architecture Enhancements DBCC TEXTALL DBCC CHECKDB DBCC TEXTALLOC DBCC CHECKTABLE DBCC DBREPAIR DROP DATABASE
<a href="#">Devices</a>	Overview of SQL Server Architecture DISK INIT CREATE DATABASE ALTER DATABASE DISK REINIT <b>sp_logdevice</b> <b>sp_dropdevice</b>

## Open Data Services

**srv\_config**  
**srv\_config\_alloc**  
**srv\_getconfig**  
**srv\_init**  
**srv\_run**  
**srv\_tdsversion**  
**srv\_getuserdata**  
**srv\_setuserdata**  
**srv\_errhandle**  
**srv\_iodead**  
**srv\_log**  
**srv\_sendstatus**  
**srv\_sfield**  
**srv\_event**  
**srv\_eventdata**  
**srv\_getserver**  
**srv\_got\_attention**  
**srv\_handle**  
**srv\_pre\_handle**  
**srv\_post\_handle**  
**srv\_setevent**  
**srv\_terminatethread**  
**srv\_attention**  
**srv\_connect**  
**srv\_disconnect**  
**srv\_language**  
**srv\_rpc**  
**srv\_exit**  
**srv\_start**  
**srv\_sleep**  
**srv\_restart**  
**srv\_stop**  
**srv\_langcpy**  
**srv\_langlen**  
**srv\_langptr**  
**srv\_paramdata**  
**srv\_paramlen**

	<b>srv_parammaxlen</b> <b>srv_paramname</b> <b>srv_paramnnumber</b> <b>srv_paramset</b> <b>srv_paramstatus</b> <b>srv_paramtype</b> <b>srv_returnval</b> <b>srv_rpcdb</b> <b>srv_rpcnumber</b> <b>srv_rpcoptions</b> <b>srv_clearstatistics</b> <b>srv_sendstatistics</b> <b>srv_alloc</b> <b>srv_bmove</b> <b>srv_bzero</b> <b>srv_free</b> <b>srv.h</b> <b>srv_describe</b> <b>srv_setcollen</b> <b>srv_setcoldata</b> <b>srv_paramsetoutput</b> <b>srv_paraminfo</b>
<a href="#">Query Performance</a>	<b>SUSER_ID</b> <b>SUSER_SID</b> <b>SUSER_NAME</b> <b>SUSER_SNAME</b> <b>syslogins</b> <b>sysdatabases</b> <b>sysremotelogins</b> <b>sysusers</b> <b>sysalternates</b>
<a href="#">Security</a>	<b>GRANT</b> <b>Authentication</b> <b>SETUSER</b>
<a href="#">SELECT</a>	<b>FASTFIRSTROW</b> <b>SELECT</b> <b>INDEX = (index hint)</b>

<a href="#">SET SHOWPLAN</a>	<b>SET SHOWPLAN_TEXT</b> <b>SET SHOWPLAN_ALL</b> <b>SQLGetDiagRec</b>
<a href="#">System Stored Procedures (Extended)</a>	<b>xp_grantlogin</b> <b>xp_revokelogin</b> <b>sp_grantlogin</b> <b>sp_revokelogin</b>
<a href="#">System Stored Procedures (System)</a>	<b>sp_add_job</b> <b>sp_add_jobschedule</b> <b>sp_add_jobstep</b> <b>sp_addtask</b> <b>sp_delete_job</b> <b>sp_delete_jobschedule</b> <b>sp_delete_jobstep</b> <b>sp_droptask</b> <b>sp_help_jobhistory</b> <b>sp_help_jobschedule</b> <b>sp_help_jobstep</b> <b>sp_helptask</b> <b>sp_purge_jobhistory</b> <b>sp_reassigntask</b> <b>sp_start_job</b> <b>sp_stop_job</b> <b>sp_update_job</b> <b>sp_update_jobschedule</b> <b>sp_update_jobstep</b> <b>sp_updatetask</b>

## Level 4: Handling Minor Changes to Behavior

Backward Compatibility Level 4 consists of minor changes in Microsoft® SQL Server™ 2000 that produce different behavior from earlier versions of SQL Server. For example, items in this level are either ignored or have one or more ignored parameters, changes to byte lengths, added parameters or columns, or changed data type columns.

This topic covers backward compatibility details for these items.

This subheading	Relates to these items
<a href="#">Aliases</a>	Roles Managing Permissions
<a href="#">Backup and Restore</a>	RESTORE HEADERONLY LOAD HEADERONLY
<a href="#">Configuration</a>	<b>sp_configure (media retention option)</b> Setting Configuration Options
<a href="#">CREATE PROCEDURE</a>	CREATE TABLE SELECT INTO
<a href="#">Data Types</a>	<b>decimal</b> and <b>numeric</b> Using Mathematical Functions + (Add) - (Subtract) * (Multiply) / (Divide) ATN2 AVG CAST and CONVERT EXP POWER RADIANS ROUND SUM
<a href="#">DATEPART and SET</a>	SET DATEFIRST

<a href="#">DATEFIRST</a>	DATEPART
<a href="#">DBCC</a>	DBCC
<a href="#">DBCS String Comparisons</a>	Unicode space characters
<a href="#">DELETE and SELECT</a>	FROM
<a href="#">Devices</a>	ALTER DATABASE
<a href="#">Functions</a>	@@DBTS
<a href="#">Global Variables</a>	Functions
<a href="#">ODBC</a>	SQL_COPT_SS_PERF_QUERY_INTERVAL <b>SQLMoreResults</b> SQL_NO_DATA
<a href="#">Rebuilding the master Database</a>	Rebuild Master Utility
<a href="#">Rebuilding the Registry (Level 4)</a>	setup/t RegistryRebuild = On
<a href="#">Replication</a>	Replication Between Different Versions of SQL Server Subscribing to One or More Articles of a Publication
<a href="#">Security</a>	SYSTEM_USER
<a href="#">SELECT</a>	SELECT FROM
<a href="#">Triggers and System Stored Procedures (System)</a>	CREATE TRIGGER <b>sp_dboption (recursive triggers option)</b> <b>sp_tableoption</b> <b>xp_readmail</b> <b>xp_sendamil</b>
<a href="#">UPDATE</a>	UPDATE
<a href="#">Utilities</a>	SQL Query Analyzer <b>isql</b> utility

## Installing SQL Server

## After Installing or Upgrading to SQL Server 2000

For a standard installation, components include:

- The SQL Server relational database engine.
- System databases used to store system level information such as login and configuration settings and for use as database templates.
- The **pubs** and **Northwind** sample databases, provided as learning tools.
- Stored procedures, a recompiled collection of Transact-SQL statements.
- Interactive management tools used for administering SQL Server.
- SQL Server Books Online, the complete documentation for SQL Server 2000.

### See Also

[Books Online](#)

[Management Tools](#)

[Select Components](#)

[SQL Stored Procedures](#)

[Using the Start Menu](#)

[System and Sample Databases](#)

[System Databases and Data](#)

# Installing SQL Server

## Using the Start Menu

Microsoft® SQL Server™ 2000 Setup creates the **Microsoft SQL Server** program group on the **Start** menu in the **Programs** group. From the **Microsoft SQL Server** program group, you can start:

- Books Online.
- Client Network Utility.
- Configure SQL XML Support in IIS.
- Enterprise Manager.
- Import and Export Data.
- Profiler.
- Query Analyzer.
- Server Network Utility.
- Service Manager.

## Microsoft SQL Server-Switch Program Group

If you install an instance of Microsoft SQL Server 2000 (default or named) on the same computer with an installation of Microsoft SQL Server version 6.5, Setup removes the **Microsoft SQL Server 6.5** program group and adds the **Microsoft SQL Server-Switch** program group. SQL Server 6.5 and SQL Server 2000 cannot run at the same time, but using the **Server-Switch** program group,

you can switch between the two versions.

On the **Start** menu, only the program group of the active version of SQL Server is accessible at any given time. The nonactive version is shown in the **Server-Switch** program group so you can quickly switch from one version to another.

The **Microsoft SQL Server-Switch** program group contains these options:

- Microsoft SQL Server 6.5 or Microsoft SQL Server 2000 (the nonactive version)
- SQL Server Upgrade Wizard
- Uninstall SQL Server 6.5

## **See Also**

[Switching Between SQL Server 6.5 and SQL Server 2000](#)

# Installing SQL Server

## System and Sample Databases

When Microsoft® SQL Server™ 2000 is installed, Setup creates the database and log files shown in this table.

Database	Database file	Log file
<b>master</b>	Master.mdf	Mastlog.ldf
<b>model</b>	Model.mdf	Modellog.ldf
<b>msdb</b>	Msdbdata.mdf	Msdblog.ldf
<b>tempdb</b>	Tempdb.mdf	Templog.ldf
<b>pubs</b>	Pubs.mdf	Pubs_log.ldf
<b>Northwind</b>	Northwnd.mdf	Northwnd.ldf

The system databases are **master**, **model**, **msdb**, and **tempdb**. The sample databases, **pubs** and **Northwind**, are provided as learning tools. (Names of these databases are case-sensitive.) Many of the examples in SQL Server Books Online are based on the sample databases.

**Note** The default location of the database and log files is Program Files\Microsoft SQL Server\Mssql\Data. This location may vary if the default location was changed when SQL Server was installed.

### See Also

[Northwind Sample Database](#)

[pubs Sample Database](#)

[System Databases and Data](#)

# Installing SQL Server

## Locating Directories and Files

The following tables and illustration show the default locations of directories and files for Microsoft® SQL Server™ 2000 (primarily for the default instance). Depending on the options you install, all of the files listed in the tables may not appear on your computer, and others not listed may be included.

Paths listed here are default paths, and may vary if locations were changed during installation. Both program and data file locations can be changed, but the location of shared tools cannot be changed.

**IMPORTANT** Do not delete any of the following directories or their contents: Binn, Data, Ftdata, HTML, or 1033. You may delete other directories, if necessary; however, you may not be able to retrieve any lost functionality or data without uninstalling and reinstalling SQL Server 2000.

Do not delete or modify any of the .htm files in the HTML directory. They are required for SQL Server Enterprise Manager and other tools to function properly.

### Shared Files for All Instances of SQL Server 2000

This table shows the locations for the shared files for both default and named instances of SQL Server 2000.

Location	Description
\Program Files\Microsoft SQL Server\80\Com	Dynamic-link libraries (DLLs) for Component Object Model (COM) objects.
\Program Files\Microsoft SQL Server\80\Com\Binn\Resources\1033	Resource files (RLLs) used by the DLLs in this COM directory. (Note: 1033 is for U.S. English; localized versions use different directory numbers.)
\Program Files\Microsoft SQL Server\80\Tools\Binn	Microsoft Windows NT® client executable files.
\Program Files\Microsoft SQL	Resource files used by the DLLs in

Server\80\Tools\Binn\Resources\1033	the Tools\Binn directory.
\Program Files\Microsoft SQL Server\80\Tools\Books	SQL Server Books Online files, including online Help files.
Program Files\Microsoft SQL Server\80\Tools\DevTools\	Header files, library files, and sample programs for use by developers.
Program Files\Microsoft SQL Server\80\Tools\Html	Microsoft Management Console (MMC) and SQL Server HTML files.
Program Files\Microsoft SQL Server\80\Tools\Templates	Boilerplate files with SQL scripts to help you create objects in the database.

## Program and Data Files for the Default Instance of SQL Server 2000

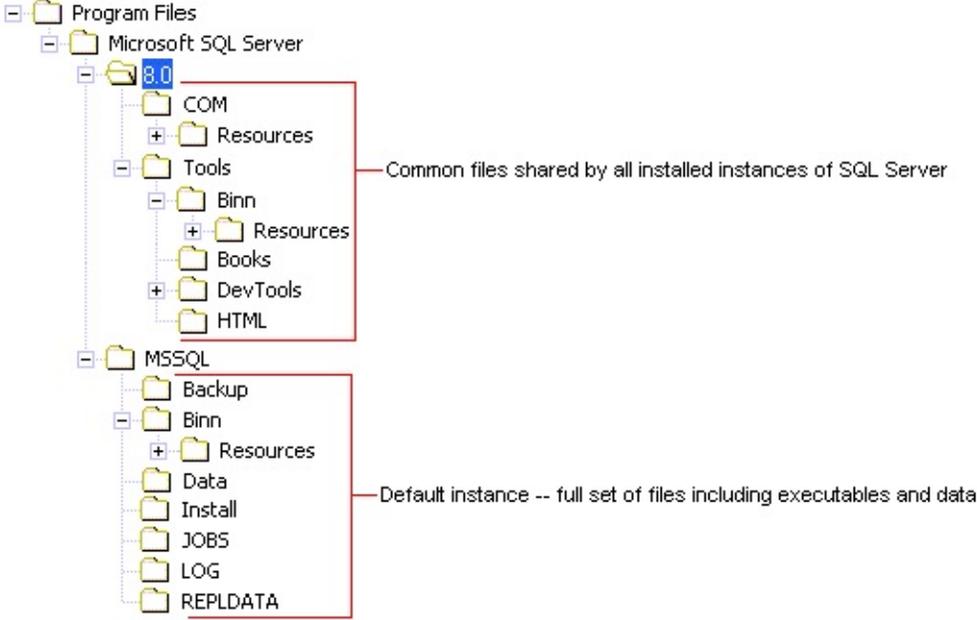
This table shows the locations of the program and data files for the default instance of SQL Server 2000. These are the default file locations, which can be changed during installation.

Location	Description
\Program Files\Microsoft SQL Server\Mssql\Backup	Default location for backup files.
\Program Files\Microsoft SQL Server\Mssql\Binn	Microsoft Windows NT server executable files and DLL files for extended stored procedures.
\Program Files\Microsoft SQL Server\Mssql\Binn\Resources\1033	Resource files used by the DLLs in this Binn directory.
Program Files\Microsoft SQL Server\Mssql\Data	System and sample database files.
Program Files\Microsoft SQL Server\Mssql\Ftdata	Full-text catalog files.
Program Files\Microsoft SQL Server\Mssql\Install	Scripts run during Setup and resulting output files.
Program Files\Microsoft SQL Server\Mssql\Jobs	Storage location for temporary job output files.

Program Files\Microsoft SQL Server\Mssql\Log	Error log files.
Program Files\Microsoft SQL Server\Mssql\Repldata	Working directory for replication tasks.
Program Files\Microsoft SQL Server\Mssql\Upgrade	Files used for version upgrade from SQL Server version 6.5 to SQL Server 2000.

### File Locations for the Default Instance of SQL Server 2000

This illustration shows the file locations for the default instance of Microsoft® SQL Server™ 2000.



### See Also

[File Paths for SQL Server 2000](#)

# Installing SQL Server

## Changing Passwords and User Accounts

Microsoft® SQL Server™ 2000 services accounts and passwords are linked to Microsoft Windows® user accounts and passwords. Changes in one location may require changes in the other.

### Changing SQL Server Services Accounts After Install

After you have installed SQL Server 2000, use SQL Server Enterprise Manager to change the assigned password or other properties of any SQL Server–related service. Each service must be changed individually. The new user account takes effect when the service is restarted. You should not change the passwords for any of the SQL Server service accounts when a failover cluster node is down or offline. If you have to do this, you will need to reset the password again using Enterprise Manager when all nodes are back online.

If you are running Microsoft Windows NT®, and you select to change the current service account for SQL Server to a non-administrator account (and the current service account for SQL Server is not an administrator account), the Valid Administrator Login dialog box is displayed. SQL Server must have administrator privileges to change security entries, so you must enter the user name, password, and domain to impersonate the non-administrator service account you have selected.

Once you have specified this information, all objects are granted full control permission. The location of the objects is determined by the following:

- Permissions are set for all files in the binary and data installation locations for the specific instances.
- Registry permissions depend on whether the instance is default or named:

For a default instance, permissions are applied only to the entries listed below the HKLM\Software\Microsoft\MSSQLServer entry:

- SQLServerAgent

- Replication
- Providers
- Setup
- Tracking
- MSSQLServer

For a named instance, permissions are applied to the entire HKLM\Software\Microsoft\MicrosoftSQLServer\80 entry.

The following rights are granted to the accounts:

- SeServiceLogonRight, which allows the account to run as a service.
- SeLockMemoryPrivilege, which allows the account to use the AWE memory feature of SQL Server.
- SeTcbPrivilege, which allows the account to impersonate other accounts.

If you are running SQL Server in a failover cluster configuration, permissions are also set for all files in the binary and data installation locations for all nodes in the cluster. Permission is also granted for the service account on the Cluster Object.

**Note** If you are running Microsoft Windows 2000 and want to use the Windows 2000 Encrypted File System to encrypt any SQL Server files, you must unencrypt the files before you can change the SQL Server service accounts. If you do not unencrypt the files and then reset the SQL Server service accounts, you cannot unencrypt the files.

You can change the SQLServerAgent service account to a non Microsoft Windows NT® 4.0 administrator account. However, the Windows NT 4.0

account must be a member of the **sysadmin** fixed server role to run SQL Server Agent.

**To change the MSSQLServer services login (Enterprise Manager)**

# Installing SQL Server

## Renaming a Server

When you change the name of the computer that is running Microsoft® SQL Server™ 2000, the new name is recognized during SQL Server startup. You do not have to run Setup again to reset the computer name.

You can connect to SQL Server using the new computer name after you have restarted the server. However, to correct the **sys.servers** system table, you should manually run these procedures:

```
sp_dropserver <old_name>  
go  
sp_addserver <new_name>  
go
```

## Issues with Remote Logins and Replication

If the computer has any remote logins, for example, if it is a replication Publisher or Distributor, **sp\_dropserver** may generate an error similar to this:

```
Server: Msg 15190, Level 16, State 1, Procedure sp_dropserver, Line 4  
There are still remote logins for the server 'SERVER1'.
```

To resolve the error, you may need to drop remote logins for this server. If replication is installed, disable replication on the server before running the **sp\_dropserver** stored procedure.

### To disable replication using the SQL Server Enterprise Manager

1. Expand a server group, and then expand the Distributor (the server that contains the distribution database).
2. Right-click the Replication folder, and then click **Disable Publishing**.
3. Complete the steps in the Disable Publishing and Distribution Wizard.

# Installing SQL Server

## **Deploying SQL Server After Initial Installation**

Microsoft® SQL Server™ 2000 includes a new method for distributing a disk image of an installation. When an installation is first created, it is marked as a new installation. When the server is restarted after installation, SQL Server 2000 verifies that the server name has not changed. If the server name has changed, an automatic correction is made.

This functionality allows Independent Service Vendors to install SQL Server 2000, stop the server, clone the disk image, and then distribute it as required. On the first startup of the distributed server, the name correction is made.

This process can be done only one time. If the server is restarted and then stopped, a new SQL Server installation must be created to be distributed as an image during deployment.

# Installing SQL Server

## Installing Full-Text Search and Indexing Tools

The full-text search engine (Microsoft Search service) is installed by default with a typical installation of Microsoft® SQL Server™ 2000, Standard and Enterprise editions.

**IMPORTANT** If upgrading from SQL Server 7.0 to SQL Server 2000 and full-text search is not installed in SQL Server 7.0, install full-text search as an additional component after the upgrade is completed.

### Microsoft Indexing Service Version 2.0

In addition to using full-text search on character columns in SQL Server data, you can use Microsoft Indexing Service along with Microsoft Search service to make textual queries against data residing in the file system. This indexing service is included in Microsoft Windows® 2000. Microsoft Windows NT® users can install the indexing service from the Microsoft Windows NT 4.0 Option Pack.

#### To install Microsoft Indexing Service 2.0 (Windows NT only)

1. Install Windows NT 4.0 Option Pack.
2. Select **Index Server 2.0**. An error appears stating that Index Server 2.0 did not install properly.
3. Install Windows NT 4.0 Service Pack 4.
4. Reinstall Windows NT 4.0 Option Pack. You are not required to change any settings.
5. Optional step: Repeat the reinstallation of Windows NT 4.0 Service Pack 4.

Repeating this procedure ensures proper installation.

## **See Also**

[Full-text Querying of File Data](#)

[How to add components to an instance of SQL Server 2000 \(Setup\)](#)

# Installing SQL Server

## **Configuring SQL Server 2000 After Upgrading**

After the server is upgraded to Microsoft® SQL Server™ 2000, you may want to perform several configuration tasks, for example:

- Set server configuration parameters.
- Set security parameters.
- Register the server and add it to a server group.

Use SQL Server Enterprise Manager or Transact-SQL to perform these tasks.

### **See Also**

[Administering SQL Server Overview](#)

# Installing SQL Server

## Switching Between SQL Server 6.5 and SQL Server 2000

Microsoft® SQL Server™ 2000 can be installed on the same computer with Microsoft SQL Server version 6.5, but only one version can be active at one time.

**Note** Switching between versions is not the same as running multiple instances. For more information, see [Working with Instances and Versions of SQL Server](#).

After the SQL Server Upgrade Wizard finishes the conversion from SQL Server version 6.5, SQL Server 2000 is the active version of SQL Server. If enough disk space exists on your computer, you can keep the SQL Server 6.5 installation intact.

To switch from one version to the other, use the **Microsoft SQL Server-Switch** application on the **Start** menu, or run Vswitch.exe.

**IMPORTANT** Be sure the SQL Server Upgrade Wizard is finished with its upgrade tasks, before you attempt to switch between the active and nonactive versions of SQL Server.

**To switch from SQL Server 6.5 to SQL Server 2000**

# Installing SQL Server

## Removing SQL Server 7.0 or SQL Server 6.5 After Upgrading

After you upgrade your Microsoft® SQL Server™ version 7.0 installation to Microsoft SQL Server 2000, or after you upgrade your databases from SQL Server version 6.5 to SQL Server 2000, you can keep the earlier versions on your computer.

A number of configurations are available for keeping earlier versions of SQL Server alongside SQL Server 2000. For more information, see [Working with Instances and Versions of SQL Server](#).

When you are ready to remove an earlier version, you can use the Add/Remove Programs option in Control Panel, or you can run uninstall from the **Start** menu.

### To remove SQL Server 7.0

- On the **Start** menu, in the SQL Server 7.0 group, click **Uninstall SQL Server 7.0**.

### To remove SQL Server 6.5

- On the **Start** menu, in the SQL Server 6.5 group, click **Remove SQL Server 6.5**.

# Installing SQL Server

## Removing SQL Server 2000

You can remove instances of Microsoft® SQL Server™ 2000 by:

- Running SQL Server 2000 Setup and selecting the Uninstall option.
- Running the Add/Remove Programs application in Control Panel.

Each named instance of SQL Server 2000 must be removed separately. You cannot remove individual components of SQL Server 2000. To remove components, you must remove the entire instance.

**IMPORTANT** Before removing SQL Server 2000, quit all applications, including the Windows NT Event Viewer, the Registry editor, all SQL Server applications, and all applications dependent on SQL Server.

**To remove SQL Server using Control Panel**

## Installing SQL Server

### Backup and Restore (Level 1)

Because backups are not compatible between servers running Microsoft® SQL Server™ 2000 and servers running earlier versions of SQL Server, SQL Server 6.x database dumps (backups) cannot be restored onto a SQL Server 2000 server. For more information about upgrading your databases to SQL Server 2000, see [Upgrading Databases from SQL Server 6.5 \(Upgrade Wizard\)](#).

SQL Server 6.x	SQL Server 2000
The VOLUME clause of the DUMP and LOAD statements indicated the volume ID for a dump device.	<p>The VOLUME keyword has been replaced by the MEDIANAME clause. Use of the VOLUME clause results in an error.</p> <p>Remove all references of the VOLUME keyword in all <a href="#">BACKUP</a>, <a href="#">DUMP</a>, <a href="#">LOAD</a>, or <a href="#">RESTORE</a> statements and replace with references to MEDIANAME.</p>
The DUMP and LOAD statements supported the use of diskettes.	<p>Backing up to diskette is not supported.</p> <p>Back up to hard disk, and then copy the backup file to one or more diskettes.</p>
The <b>sysbackuphistory</b> , <b>sysbackupdetail</b> , <b>sysrestorehistory</b> , and <b>sysrestoredetail</b> system tables tracked DUMP and LOAD history information.	<p>The DUMP and LOAD history tracking system tables have been removed and replaced by a new set of system tables.</p> <p>Remove all references to <b>sysbackuphistory</b>, <b>sysbackupdetail</b>, <b>sysrestorehistory</b>, and <b>sysrestoredetail</b>. Because the structure and contents of the backup system tables have changed significantly, familiarize yourself with these new system tables before referencing them:</p>

[backupfile](#), [backupmediafamily](#),  
[backupmediaset](#), [backupset](#),  
[restorefile](#), [restorefilegroup](#), and  
[restorehistory](#).

## Installing SQL Server

### Configuration Options (Level 1)

Administrative scripts may have used these configuration options. For more information about configuration options, see [Setting Configuration Options](#).

SQL Server 6.x	SQL Server 2000
<b>backup buffer size</b> specified the size of the dump and load buffer (used to increase backup speed).	Removed; no longer supported. Remove all references to <b>backup buffer size</b> .
<b>backup threads</b> specified the number of threads to be reserved for striped dump and load operations.	Removed; no longer supported. Remove all references to <b>backup threads</b> .
<b>database size</b> set the default number of megabytes (MB) allocated to each new user database.	Removed; no longer supported. Remove all references to <b>database size</b> .
<b>free buffers</b> determined the threshold of free buffers available to the system.	Removed; no longer supported. Remove all references to <b>free buffers</b> .
<b>hash buckets</b> set the number of buckets used for hashing pages to buffers in memory.	Removed; no longer supported. Remove all references to <b>hash buckets</b> .
<b>LE threshold maximum</b> determined the maximum number of page locks to hold before escalating to a table lock.	Removed; no longer supported. Remove all references to <b>LE threshold maximum</b> .
<b>LE threshold minimum</b> determined the minimum number of page locks required before escalating to a table lock.	Removed; no longer supported. Remove all references to <b>LE threshold minimum</b> .
<b>LE threshold percent</b> specified the percentage of page locks needed on a table before a table lock is requested.	Removed; no longer supported. Remove all references to <b>LE threshold percent</b> .
<b>logwrite sleep</b> specified the number of milliseconds that a write to the log	Removed; no longer supported. Remove all references to <b>logwrite</b>

will be delayed if the buffer is not full.	<b>sleep.</b>
<b>max lazywrite IO</b> tuned the priority of batched asynchronous I/O operations performed by the lazy writer.	Removed; no longer supported. Remove all references to <b>max lazywrite IO</b> .
<b>memory</b> set the size of available memory, in 2K units.	Removed; no longer supported. Memory is configured automatically based on need and available memory. To control the range of memory configured automatically, use the <b>min server memory</b> and <b>max server memory</b> options. Remove all references to <b>memory</b> .
<b>open databases</b> set the maximum number of databases that can be open at one time on SQL Server.	Removed; no longer supported. Remove all references to <b>open databases</b> .
<b>procedure cache</b> specified the percentage of memory allocated to the procedure cache after the SQL Server memory needs are met.	Removed; no longer supported. Remove all references to <b>procedure cache</b> .
<b>RA cache hit limit</b> specified the number of cache hits that a read-ahead request could have before it was canceled.	Removed; no longer supported. Remove all references to <b>RA cache hit limit</b> .
<b>RA cache miss limit</b> specified the number of cache misses that occurred during a horizontal traversal before read-ahead started for that command.	Removed; no longer supported. Remove all references to <b>RA cache miss limit</b> .
<b>RA delay</b> specified the delay of read-ahead, in milliseconds.	Removed; no longer supported. Remove all references to <b>RA delay</b> .
<b>RA pre-fetches</b> determined how far ahead the read-ahead manager read (on an extent basis) before the pre-fetch manager idled.	Removed; no longer supported. Remove all references to <b>RA pre-fetches</b> .

<p><b>RA slots per thread</b> specified the number of simultaneous requests each read-ahead service thread managed.</p>	<p>Removed; no longer supported. Remove all references to <b>RA slots per thread</b>.</p>
<p><b>RA worker threads</b> specified the number of threads used to service read-ahead requests.</p>	<p>Removed; no longer supported. Remove all references to <b>RA worker threads</b>.</p>
<p><b>recovery flags</b> determined what information SQL Server displayed in the error log during recovery.</p>	<p>Removed; no longer supported. Remove all references to <b>recovery flags</b>.</p>
<p><b>remote conn timeout</b> specified a time limit to break a server-to-server connection.</p>	<p>Removed; no longer supported. Remove all references to <b>remote conn timeout</b>.</p>
<p><b>SMP concurrency</b> specified the maximum number of CPUs that would be used by SQL Server.</p>	<p>Removed; no longer supported. Remove all references to <b>SMP concurrency</b>.</p>
<p><b>sort pages</b> specified the maximum number of pages to be allocated to sorting per user.</p>	<p>Replaced by <b>min memory per query</b>. For more information about the <b>min memory per query</b> option, see <a href="#">Server Memory Options</a>.</p> <p>Replace all references of <b>sort pages</b> with <b>min memory per query</b> and <b>index create memory</b>.</p>
<p><b>tempdb in ram</b> placed the <b>tempdb</b> database in RAM, if needed.</p>	<p>No longer supported because SQL Server 2000 has been optimized for maximum performance.</p> <p>Remove all references to <b>tempdb in ram</b>.</p>
<p><b>trace flag 204</b> supported queries containing sort columns in the ORDER BY clause not included in the <i>select list</i> when the DISTINCT keyword was supplied.</p>	<p>No longer supported. Remove all references to <b>trace flag 204</b>. For more information about supported trace flags, see <a href="#">Trace Flags</a>.</p>
<p><b>user connections</b> set the maximum</p>	<p>Now an advanced option. Default</p>

number of simultaneous connections to SQL Server allowed.	value of 0 indicates automatic growth. Remove all references to <b>user connections</b> .
---	---

For more information about other changes to configuration options, see SQL Server 2000 and SQL Server version 7.0.

## Installing SQL Server

### Custom Sort Orders (Level 1)

SQL Server 6.x	SQL Server 2000
Custom sort orders were installed from definition files (usually with an .srt file extension).	Removed; no longer available or supported. Remove all references to custom sort orders. During installation of SQL Server 2000, select an appropriate sort order. For more information, see <a href="#">Windows Collation Sorting Styles</a> .

## Installing SQL Server

### Databases (Level 1)

SQL Server 6.x	SQL Server 2000
<p>The ON <i>database_device = size</i> clause of ALTER DATABASE specified the amount of space, in megabytes (MB), allocated to the database extension and could be used following DISK INIT to alter the database device size.</p>	<p>If the file was not created originally by DISK INIT, the ON <i>database_device = size</i> syntax cannot be specified with ALTER DATABASE. Instead, use the MODIFY FILE clause of ALTER DATABASE to alter the size of a database file. Remove all references of the ON <i>database_device = size</i> clause of ALTER DATABASE. For more information, see <a href="#">ALTER DATABASE</a>.</p>

## Installing SQL Server

### Database Options (Level 1)

Administrative scripts may have used these database options. In SQL Server 2000, database options should be set with the ALTER DATABASE statement rather than the **sp\_dboption** stored procedure. For more information about database options, see [Setting Database Options](#) and [ALTER DATABASE](#).

SQL Server 6.x	SQL Server 2000
The <b>subscribe</b> option of <b>sp_dboption</b> enabled or disabled a database for subscriptions.	Removed; no longer available. Use <a href="#">sp_addsubscription</a> to enable or disable a database for subscriptions.
The <b>no chkpt. on recovery</b> option of <b>sp_dboption</b> defined whether or not a checkpoint record was added to a database recovered during a SQL Server startup.	Removed; no longer available. When using a warm standby server, use the WITH STANDBY clause of the <a href="#">RESTORE</a> statement.

## Installing SQL Server

### Data Access Objects (DAO) (Level 1)

SQL Server 6.x	SQL Server 2000
Version 3.x of the Data Access Objects (DAO) functioned properly when accessing SQL Server version 6.x servers.	Because the ODBC driver that ships with SQL Server 2000 exposes new GUID and Unicode data types when connecting to SQL Server, DAO version 3.x does not work properly with SQL Server 2000. However, the <a href="#">odbccmpt Utility</a> is provided to enable SQL Server version 6.x ODBC compatibility for a DAO application.

## Installing SQL Server

### DBCC (Level 1)

SQL Server 6.x	SQL Server 2000
DBCC DBREINDEX used the SORTED_DATA and SORTED_DATA_REORG clauses. The SORTED_DATA clause eliminated the sort performed when a clustered index was created and physically reorganized the data. The SORTED_DATA_REORG clause eliminated the sort performed when a clustered index was created.	Removed; no longer supported. Remove all references to either the SORTED_DATA or the SORTED_DATA_REORG clauses of DBCC DBREINDEX and replace with references to the DROP_EXISTING clause of <a href="#">CREATE INDEX</a> .
DBCC SHRINKDB either returned the minimum size to which a database could shrink, or shrank the size of the specified database to the specified value.	Removed; no longer supported or available. Remove all references of DBCC SHRINKDB and replace with references to <a href="#">DBCC SHRINKDATABASE</a> . Consider shrinking databases automatically by using the AUTO_SHRINK option of <a href="#">ALTER DATABASE</a> .
DBCC MEMUSAGE provided detailed reports on memory use.	Removed; no longer supported or available. Remove all references of DBCC MEMUSAGE and replace with references to these Performance Monitor counters.

Performance Monitor object name	Performance Monitor counter name
<a href="#">SQL Server: Buffer Manager Object</a>	Procedure Cache Pages In Use
	Procedure Cache Size (pages)

<a href="#">SQL Server: Cache Manager Object</a>	Procedure Cache Hit Ratio
	Procedure Cache Pages
	Procedure Cache Object Counts*

\* These counters are available for various categories of cache objects including ad hoc sql, prepared sql, procedures, triggers, and so on.

## Installing SQL Server

### DB-Library (Level 1)

SQL Server 6.x	SQL Server 2000
DB-Library's two-phase commit special library managed transactions distributed across two or more servers.	The DB-Library two-phase commit is no longer supported. Use Microsoft Distributed Transaction Coordinator (MS DTC) to accomplish simultaneous updates on two servers. Remove all references to DB-Library's two-phase commit.
DB-Library applications could be developed in Microsoft® Visual Basic®.	The development libraries for DB-Library for Visual Basic are not supplied. Existing DB-Library for Visual Basic applications will run against SQL Server 2000, but must be maintained using the development libraries for SQL Server 6.5. All new Visual Basic applications written to access SQL Server should use the Visual Basic data APIs such as ActiveX Data Objects (ADO) and Remote Data Objects (RDO).

## Installing SQL Server

### **DECnet Network Library (Level 1)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
For Intel-based, MIPS-based, and Alpha AXP-based computers, server DECnet Sockets Net-Libraries provided connectivity with PATHWORKS networks by allowing clients running on VMS to connect to SQL Server.	Removed; no longer supported. Remove all references to the DECnet Sockets Net-Libraries.

## Installing SQL Server

### Disk Commands (Level 1)

SQL Server 6.x	SQL Server 2000
DISK REINIT and DISK REFIT restored usage information from system tables when a device existed (the file was present) but the entries in <b>sysusages</b> no longer existed.	Removed; no longer supported or available. Remove all references to DISK REINIT. Replace all references of DISK REFIT with references to <a href="#">ALTER DATABASE</a> , which adds and drops filegroups included in a database, and modifies the size of each database filegroup.

## Installing SQL Server

### **Disk Mirroring (Level 1)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
DISK MIRROR, DISK REMIRROR, and DISK UNMIRROR performed SQL Server disk mirroring.	No longer supported because SQL Server mirroring is no longer supported. Use Microsoft Windows NT® or hardware-based RAID. For more information, see your Windows NT or hardware documentation.

## Installing SQL Server

### Indexes (Level 1)

SQL Server 6.x	SQL Server 2000
The SORTED_DATA_REORG clause of CREATE INDEX eliminated the sort performed when a clustered index was created.	Replaced by the DROP_EXISTING clause of CREATE INDEX. Remove all references to the SORTED_DATA_REORG clause of <a href="#">CREATE INDEX</a> and replace with references to DROP_EXISTING.
The SORTED_DATA clause of CREATE INDEX eliminated the sort performed when a clustered index was created and physically reorganized the data.	Removed; no longer available. Remove all references to the SORTED_DATA clause of CREATE INDEX.
<b>bcp</b> could import an already sorted data file into a SQL Server table. Creating a clustered index on an ordered table could be optimized by using the SORTED_DATA clause of CREATE INDEX. The SORTED_DATA clause forced SQL Server not to sort or reorganize the previously ordered table.	SQL Server returns an error message stating that the SORTED_DATA clause of CREATE INDEX is ignored and no longer supported. Remove all references to the SORTED_DATA clause of CREATE INDEX. Consider creating the clustered index before using <b>bcp</b> to import the data. <b>bcp</b> uses improved index maintenance strategies to make data importation with a preexisting index faster than earlier releases and avoids resorting of data after importation.
The ALLOW_DUP_ROW and IGNORE_DUP_ROW clauses of the CREATE INDEX statement allowed data to be updated into tables with a unique index and without having to filter out duplicates first.	No longer supported. Using either ALLOW_DUP_ROW or IGNORE_DUP_ROW in the CREATE INDEX statement generates a warning message. If there is no unique clustered index and there is a need to avoid duplicate rows, create a unique

	constraint on one or more columns other than the clustering key.
--	--

## Installing SQL Server

### Open Data Services (Level 1)

In SQL Server 2000, Open Data Services, now called extended stored procedures, no longer supports gateway applications.

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
<p>The ODBC client driver for Open Data Services gateways (ODSGT32.DLL) and associated resource files were used by ODBC clients to connect to Open Data Services gateway servers.</p>	<p>Not shipped with SQL Server 2000. The SQL Server version 6.x ODSGT32.DLL and associated resource files work against an Open Data Services gateway recompiled with SQL Server version 7.0 headers and libraries. Use the SQL Server version 6.x ODBC client driver for Open Data Services (ODSGT32.DLL) and associated resource files to connect from an ODBC client to an Open Data Services gateway.</p> <p>Consider redesigning your application using Windows NT Component Services.</p>
<p>Open Data Services data structures such as SRV_CONFIG, SRV_PROC, and SRV_SERVER were exposed in the Open Data Services header file.</p>	<p>These data structures are no longer exposed, and the data structure members have changed. Applications that reference these data structures directly or their members must be changed and recompiled using the SQL Server 7.0 Open Data Services header file (srv.h) and relinked using the SQL Server 7.0 Open Data Services library file (opens60.lib). These changes should be made to avoid the possibility of server failures.</p>
<p>Earlier versions of SQL Server could make remote stored procedure calls against gateways</p>	<p>SQL Server 2000 does not support remote stored procedure calls against gateways compliant with 6.x and 4.x versions of</p>

compliant with 6.x or 4.x versions of Open Data Services.

Open Data Services. SQL Server 2000 does support remote stored procedure calls against gateways compliant with SQL Server 2000. Gateways compiled and linked with earlier versions of Open Data Services should be recompiled with SQL Server 7.0 version of Open Data Services. Consider using distributed query if your target data source has an ODBC or an OLE DB provider on Windows NT or Windows 95/98.

## Installing SQL Server

### **Program Group Tools and Utilities (Level 1)**

In Microsoft® SQL Server™ 2000, these tools have been renamed or replaced.

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
ISQL_w	SQL Query Analyzer
MS Query	N/A
SQL Client Configuration	Client Network Utility
SQL Enterprise Manager	SQL Server Enterprise Manager
SQL Help	N/A
SQL Security Manager	N/A
SQL Trace	SQL Server Profiler
SQL Performance Monitor	N/A. SQL Server performance counters are added to the Windows 2000 System Monitor or the Windows NT 4.0 Performance Monitor.
SQL Service Manager	SQL Server Service Manager
SQL Setup	N/A

## Installing SQL Server

### Replication (Level 1)

SQL Server 6.x	SQL Server 2000
Restricted publications could be created through the user interface and used in replicating data.	Restricted publications cannot be created through the user interface and are no longer supported. Remove all references to restricted publications. A replacement for restricted publications will be available in a later release. For more information, see <a href="#">Replication Overview</a> .
Publish and subscribe properties could be set using the <b>DBOption</b> object.	No longer available. Remove all references to the <b>DBOption</b> object and replace with references to the EnablePublishing property of the <a href="#">ReplicationDatabase object</a> .
The <b>repl_publisher</b> login allowed replication processes on the distributor to connect to a subscription server and replicated table schema and data to destination databases.	No longer available. Remove all references to the <b>repl_publisher</b> login and replace with references to a login in a publication access list (PAL). For more information about PALs, see <a href="#">Publication Access Lists</a> .

## Installing SQL Server

### Security (Level 1)

SQL Server 6.x	SQL Server 2000
DENY was not a reserved keyword and could be used as an object identifier.	DENY is a reserved keyword. Rename any object named DENY. Change all Transact-SQL statements and scripts referencing the object to use the new object name. If DENY is retained as an object identifier, all references to the object must use <a href="#">Delimited Identifiers</a> .

## Installing SQL Server

### Segments (Level 1)

SQL Server 6.x	SQL Server 2000
Indexes could be placed on segments using the CREATE INDEX statement.	Segments are no longer supported. However, CREATE INDEX can create an index on a filegroup. Remove all references to segments and replace with references to filegroups within a <a href="#">CREATE INDEX</a> statement.
Tables could be created on a particular segment by using the CREATE TABLE statement.	CREATE TABLE references files and filegroups instead of segments. Remove all references to segments and replace with references to files and filegroups within a <a href="#">CREATE TABLE</a> statement.
User-defined segments allowed the placement of database objects on certain devices for performance reasons.	Segments are no longer supported. Multidisk RAID devices generally provide a greater increase in performance with a lower associated administrative cost. Use filegroups for user-defined placement of data, indexes, or text. Remove all references to these segment-related system stored procedures:  <b>sp_addsegment</b> <b>sp_dropsegment</b> <b>sp_extendsegment</b> <b>sp_helpsegment</b>  Create, modify, or drop files and filegroups; and place indexes on files or filegroups using <a href="#">CREATE TABLE</a> , <a href="#">CREATE DATABASE</a> , <a href="#">ALTER DATABASE</a> , and <a href="#">CREATE INDEX</a> .

## Installing SQL Server

### Services (Level 1)

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
SQL Executive provided the SQL Server scheduling engine. SQL Executive offered extensive and varied task scheduling and alerting abilities, and was capable of handling large client/server environments.	SQL Executive tasks are now performed by SQL Server Agent. Use SQL Server Agent for scheduling purposes. For more information, see <a href="#">Configuring the SQL Server Agent Service</a> .

## Installing SQL Server

### **SET DISABLE\_DEF\_CNST\_CHK (Level 1)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
The SET DISABLE_DEF_CNST_CHK setting controlled interim constraint checking.	Removed; no longer available. Remove all references to SET DISABLE_DEF_CNST_CHK.

## Installing SQL Server

### SET SHOWPLAN (Level 1)

SQL Server 6.x	SQL Server 2000
<p>SET SHOWPLAN generated a description of the processing plan for the query and processed it immediately unless the SET NOEXEC setting was enabled.</p>	<p>SET SHOWPLAN has been replaced with SET SHOWPLAN_ALL and SET SHOWPLAN_TEXT. The SET SHOWPLAN_ALL and SET SHOWPLAN_TEXT statements return only query or statement execution plan information and do not execute the query or statement. To execute the query or statement, turn the appropriate showplan statement OFF. The query or statement will then execute.</p> <p>Remove all references to either SET SHOWPLAN ON or SET SHOWPLAN OFF and replace with references to either <a href="#">SET SHOWPLAN_ALL ON</a>, <a href="#">SET SHOWPLAN_TEXT ON</a>, SET SHOWPLAN_ALL OFF, or SET SHOWPLAN_TEXT OFF. Expect differences in behavior as compared to earlier versions of SQL Server.</p>

## Installing SQL Server

### SQL Alerter (Level 1)

SQL Server 6.x	SQL Server 2000
SQL Alerter, SQLALRTR.exe, was used to integrate the alert engine with the Windows NT Performance Monitor alerter.	Removed; no longer supported or available. Replaced by SQL Server performance condition alerts. Remove all references to SQL Alerter and replace with references to SQL Server performance condition alerts. For more information, see <a href="#">Defining Alerts</a> .

## Installing SQL Server

### SQL-DMO (Level 1)

SQL-DMO applications are administrative tools and should be updated to work with SQL Server 2000. It is recommended that code be recompiled and any error messages returned from the build process be used to track any necessary changes.

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
<p>SQL-DMO is implemented in Sqlole.dll. The SQL-DMO objects exhibit properties, methods, and events that automate administrative tasks for SQL Server version 6.5 and earlier. SQL-DMO, implemented in Sqlole.dll, cannot connect to and operate against SQL Server 2000 or SQL Server 7.0.</p>	<p>SQL-DMO is implemented in Sqldmo.dll. These SQL-DMO objects expose the properties, methods, and events that automate administrative tasks for SQL Server. They cannot be used to connect to and operate against a SQL Server version 6.5 (or earlier) server. Therefore, it is recommended that you rewrite SQL-DMO applications.</p> <p>If the application must operate against both SQL Server 2000 and version 6.5 or earlier of SQL Server, reference both Sqldmo.dll and Sqlole.dll components in the application. Develop new, separate subroutines referencing the SQL Server 2000 SQL-DMO objects from the existing subroutines.</p> <p>If the application will work against SQL Server 2000 only, rewrite existing subroutines to reference SQL Server 2000 SQL-DMO objects.</p> <p>If the application will not be used against your new SQL Server 2000 server(s), continue to use the application unchanged.</p>

---

## Installing SQL Server

### System Stored Procedures (General Extended Procedures) (Level 1)

SQL Server 6.x	SQL Server 2000
<b>xp_snmp_getstate</b> returned the state of the SQL Server Simple Network Management Protocol (SNMP) agent. <b>xp_snmp_raisetrail</b> permitted a client to define and send a trap (an SNMP alert) to an SNMP client.	Removed; no longer available. Remove all references to either <b>xp_snmp_getstate</b> or <b>xp_snmp_raisetrail</b> .

## Installing SQL Server

### System Stored Procedures (Replication) (Level 1)

SQL Server 6.x	SQL Server 2000
<b>sp_replica</b> remotely set (on a Subscriber) a <b>sysobjects</b> category bit that marked the table as a replica.	Removed; no longer supported or available. Remove all references to <b>sp_replica</b> .
<b>sp_replsync</b> acknowledged completion of a manual synchronization when used from a Subscriber.	Removed; no longer supported or available. Remove all references to <b>sp_replsync</b> .
<b>sp_helppublicationsync</b> provided information about a scheduled synchronization task for a publication.	No longer supported. An error message is returned if this stored procedure is used. Remove all references to <b>sp_helppublicationsync</b> .
<b>sp_subscribe</b> and <b>sp_unsubscribe</b> remotely added or canceled a subscription to a particular article within a publication, to a whole publication, or to all publications.	No longer supported. An error message is returned if this stored procedure is used. Remove all references to either <b>sp_subscribe</b> or <b>sp_unsubscribe</b> , or use the <a href="#">@@ERROR</a> function to test for errors.
<b>name value</b> parameter of <b>sp_changepublication</b> was used to provide the new publication name.	Removed; no longer supported or available. Remove all references to the <b>name value</b> parameter of <a href="#">sp_changepublication</a> .
<b>sp_addpublisher</b> added a Publisher at the Subscriber and added a Distribution Publisher at the Distributor.	Replaced by <b>sp_adddistpublisher</b> . Remove all references to <b>sp_addpublisher</b> and replace with references to <a href="#">sp_adddistpublisher</a> .
<b>sp_droppublisher</b> dropped a publication server.	Removed; no longer supported or available. Remove all references to <b>sp_droppublisher</b> . To drop a Publisher at a Distributor, use

	<a href="#"><b>sp_dropdistpublisher.</b></a>
<b>sp_distcounters</b> was used to query for delivered or undelivered commands as used by Performance Monitor, which no longer uses this procedure.	No longer supported. The new view <b>MSdistribution_status</b> presents much of the same information.
<b>sp_helpreplicationdb</b> was used to return information about a specified database or a list of all publication databases on the server.	Removed; no longer supported. Remove all references to <b>sp_helpreplicationdb</b> and replace with references to <a href="#"><b>sp_helpreplicationdboption.</b></a>
<b>sp_replstatus</b> updated the internal table structure for replication.	Removed; no longer supported. Remove references to <b>sp_replstatus.</b>

## Installing SQL Server

### System Stored Procedures (System) (Level 1)

SQL Server 6.x	SQL Server 2000
Several system stored procedures were used for documenting keys.	Removed; no longer supported or available. Use declarative referential integrity by implementing keys and constraints with either <a href="#">ALTER TABLE</a> or <a href="#">CREATE TABLE</a> . Remove all references to these system stored procedures and replace with references to either <a href="#">sp_help</a> or <a href="#">sp_helpconstraint</a> :  <b>sp_commonkey</b> <b>sp_dropkey</b> <b>sp_foreignkey</b> <b>sp_helpjoins</b> <b>sp_helpkey</b> <b>sp_primarykey</b>
<b>sp_placeobject</b> put future space allocations for a table or index on a particular segment.	<b>sp_placeobject</b> is no longer available because segments no longer exist. Use the ON FILEGROUP syntax of the CREATE TABLE statement to place table or index information about a separate filegroup. Remove all references of <b>sp_placeobject</b> and replace with references to the ON FILEGROUP clause of the <a href="#">CREATE TABLE</a> statement.
<b>sp_dbinstall</b> installed a database and its devices, and was used for removable media.	Removed; no longer supported or available. Remove all references to <b>sp_dbinstall</b> and replace with references to <a href="#">sp_attach_db</a> .

<p><b>sp_makestartup</b> and <b>sp_unmakestartup</b> set a stored procedure for auto execution and discontinued auto execution of the stored procedure, respectively.</p>	<p>Removed; no longer supported or available. Remove all references of either <b>sp_makestartup</b> or <b>sp_unmakestartup</b> and replace with references to <a href="#">sp_procoption</a>.</p>
<p>The <b>sp_helplogins</b>, <b>sp_helpprotect</b>, and <b>sp_tableoption</b> system stored procedures supported pattern matching (using wildcard characters), which allowed flexibility in specific parameters.</p>	<p>Pattern matching using the wildcard characters is no longer supported in these system stored procedures because any system stored procedure identifier may contain a pattern matching character. Remove all references to pattern matching in <a href="#">sp_helplogins</a>, <a href="#">sp_helpprotect</a>, and <a href="#">sp_tableoption</a>.</p>
<p>The <b>fallback</b> option of <b>sp_serveroption</b> indicated a fallback server.</p>	<p>The <b>fallback</b> option of <b>sp_serveroption</b> is no longer available because the <b>fallback</b> option is no longer supported. Remove all references to the <b>fallback</b> option of <a href="#">sp_serveroption</a>.</p>
<p><b>sp_setlangalias</b> assigned or changed the alias for an alternate language.</p>	<p>Removed; no longer supported. Use the aliases provided in <b>syslanguages</b>. Remove all references to <b>sp_setlangalias</b>.</p>
<p><b>sp_droplanguage</b> dropped an alternate language from the server and removed its row from <b>master.dbo.syslogins</b>.</p>	<p>Removed; no longer supported. Remove all references to <b>sp_droplanguage</b>.</p>
<p>Fallback support was provided by executing system stored procedures that shifted control of databases and devices from a broken primary server to a fallback server.</p>	<p>Fallback support is no longer supported using the fallback system stored procedures. Support for fallback servers is supported using Microsoft Windows NT Clustering Service. Remove all references to these fallback system stored procedures:</p>

	<p><b>sp_fallback_activate_svr_db,</b>  <b>sp_fallback_deactivate_svr_db,</b>  <b>sp_fallback_enroll_svr_db,</b>  <b>sp_fallback_help,</b>  <b>sp_fallback_permanent_svr,</b>  <b>sp_fallback_upd_dev_drive,</b>  <b>sp_fallback_withdraw_svr_db.</b></p>
<p><b>sp_devoption</b> displayed or set device status.</p>	<p>Removed; no longer available.  Remove all references to <b>sp_devoption.</b></p>
<p><b>sp_diskdefault</b> set a database device status to indicate whether the device can be used for database storage when the user does not specify a database device or specifies DEFAULT with the CREATE DATABASE or ALTER DATABASE statements.</p>	<p>Removed; no longer available.  Remove all references to <b>sp_diskdefault.</b></p>
<p><b>sp_helplog</b> reported the name of the device that contains the first page of the log in the current database.</p>	<p>Removed; no longer available.  Remove all references to <b>sp_helplog.</b></p>
<p><b>sp_helpstartup</b> reported a listing of all auto-start stored procedures.</p>	<p>Removed; no longer available.  Remove all references to <b>sp_helpstartup</b> and replace with references to <b>sp_procoption.</b></p>
<p><b>sp_sqlexec</b> provided a convenient way for SQL Server database clients and servers to send a language statement of any format to an Open Data Services server application.</p>	<p>Removed; no longer available.  Remove all references to <b>sp_sqlexec.</b></p>
<p><b>sp_helprevedatabase</b> analyzed an existing database and created a script that could be used to replicate the database structure on another server.</p>	<p>Removed; no longer available. If applicable, use the SQL-DMO <a href="#">Script Method</a> of the <a href="#">Database Object</a> to generate similar information.  Remove all references to <b>sp_helprevedatabase.</b></p>

**sp\_addlanguage** added an alternate language to a server.

Removed; no longer available.  
Remove all references to **sp\_addlanguage**.

## Installing SQL Server

### System Stored Procedures (Tasks) (Level 1)

Replace the following unsupported Microsoft® SQL Server™ 6.x task-related system stored procedures with the corresponding SQL Server 2000 job-related system stored procedures.

SQL Server 6.x	SQL Server 2000
<b>sp_addalert</b>	<a href="#"><u>sp_add_alert</u></a>
<b>sp_addnotification</b>	<a href="#"><u>sp_add_notification</u></a>
<b>sp_addoperator</b>	<a href="#"><u>sp_add_operator</u></a>
<b>sp_dropalert</b>	<a href="#"><u>sp_delete_alert</u></a>
<b>sp_dropnotification</b>	<a href="#"><u>sp_delete_notification</u></a>
<b>sp_dropoperator</b>	<a href="#"><u>sp_delete_operator</u></a>
<b>sp_helpalert</b>	<a href="#"><u>sp_help_alert</u></a>
<b>sp_helphistory</b>	<a href="#"><u>sp_help_jobhistory</u></a>
<b>sp_helpnotification</b>	<a href="#"><u>sp_help_notification</u></a>
<b>sp_helpoperator</b>	<a href="#"><u>sp_help_operator</u></a>
<b>sp_purgehistory</b>	<a href="#"><u>sp_purge_jobhistory</u></a>
<b>sp_runtask</b>	<a href="#"><u>sp_start_job</u></a>
<b>sp_stoptask</b>	<a href="#"><u>sp_stop_job</u></a>
<b>sp_updatealert</b>	<a href="#"><u>sp_update_alert</u></a>
<b>sp_updatenotification</b>	<a href="#"><u>sp_update_notification</u></a>
<b>sp_updateoperator</b>	<a href="#"><u>sp_update_operator</u></a>

Task management has been changed to job management.

## Installing SQL Server

### System Tables (Level 1)

SQL Server 6.x	SQL Server 2000
System tables were used internally by SQL Server for a wide range of uses, including maintaining the list of character sets that SQL Server could use and containing information about active locks.	System tables have changed significantly. Most SQL Server 6.x system tables will continue to work properly. Views provided allow applications referencing SQL Server 6.x system tables to continue functioning properly. However, some SQL Server 2000 data cannot be referenced through these views. Use the provided <a href="#">Information Schema Views</a> or ODBC catalog <a href="#">system stored procedures</a> to obtain system table information. Modify scripts as appropriate. Any scripts referencing SQL Server 6.x system tables will not be converted properly.
<b>sysdevices</b> contained one row for each disk dump, tape dump, and database device.	The <b>mirrorname</b> and <b>stripeset</b> columns have been removed. <b>sysdevices</b> is retained only for dump devices and also for backward compatibility (supporting DISK INIT and SQL Server 6.x CREATE DATABASE syntax). Remove all references to the <b>mirrorname</b> and <b>stripeset</b> columns of <a href="#">sysdevices</a> .
<b>syshistory</b> contained one row for each scheduled event, alert, or task that occurred.	Replaced by <b>sysjobhistory</b> . Remove all references to <b>syshistory</b> and replace with references to <a href="#">sysjobhistory</a> .
<b>sysindexes</b> contained one row for each clustered index and one row for each nonclustered index.	The <b>distribution</b> , <b>segment</b> , <b>rowpage</b> , <b>keys1</b> , and <b>keys2</b> columns have been removed. Remove all references to the <b>distribution</b> , <b>segment</b> , <b>rowpage</b> , <b>keys1</b> , and <b>keys2</b> columns of <a href="#">sysindexes</a> .  In addition, <b>soid</b> is <b>reserved3</b> , and <b>csid</b> is

	<b>reserved4.</b>
<b>syskeys</b> used for objects	Removed; no replacement. Remove all references to <b>syskeys</b> .
<b>syslocks</b> contained information about active locks.	Removed; replaced by <b>syslockinfo</b> . Remove all references to <b>syslocks</b> and replace with references to <a href="#">syslockinfo</a> .
<b>syslogs</b> contained the transaction log.	Removed; no replacement. The database log is now an operating system file. Remove all references to <b>syslogs</b> .
<b>sysprocesses</b> contained information about SQL Server processes.	The <b>gid</b> column has been removed. In SQL Server, the <b>suid</b> column has been removed. Remove all references to these columns of <a href="#">sysprocesses</a> .
<b>sysprocedures</b> contained entries for each view, default, rule, trigger, CHECK constraint, DEFAULT constraint, and stored procedure.	Removed; replaced by <b>syscomments</b> . SQL Server obtains procedure text from <b>syscomments</b> when procedures need to be compiled. Remove all references to <b>sysprocedures</b> and replace with references to <a href="#">syscomments</a> .
<b>syssegments</b> contained one row for each segment (named collection of disk fragments).	Removed; no replacement. Segments are no longer supported. Use filegroups instead. Remove all references to <b>syssegments</b> . Use filegroups instead by using <a href="#">CREATE DATABASE</a> , <a href="#">ALTER DATABASE</a> , <a href="#">CREATE TABLE</a> , <a href="#">ALTER TABLE</a> , and <a href="#">CREATE INDEX</a> .
<b>systasks</b> contained one row for every scheduled task.	Removed; replaced by <b>sysjobs</b> , <b>sysjobsteps</b> , and <b>sysjobservers</b> . Remove all references to <b>systasks</b> and replace with references to <a href="#">sysjobs</a> , <a href="#">sysjobsteps</a> , and <a href="#">sysjobservers</a> as appropriate.
<b>sysusages</b> contained one row for each disk-allocation piece assigned to a database.	Removed; no replacement. SQL Server relies on <b>sysdevices</b> for database file information. Filegroups are supported, and the <b>sysfiles</b> and <b>sysfilegroups</b> system tables are added. These system tables

	reside in every database and describe database files and filegroups. Remove all references to <b>sysusages</b> .
<b>master.dbo.spt_datatype_info</b>	Removed; no replacement. Remove all references to <b>master.dbo.spt_datatype_info</b> .

For more information, see [System Tables](#).

## Installing SQL Server

### Transactions (Level 1)

SQL Server 6.x	SQL Server 2000
<p>A data type conversion error not inside a transaction returned an error to SQL Server. If the data type conversion was inside a transaction, the transaction continued. For example:</p> <pre>USE pubs CREATE TABLE test (c1 int) GO BEGIN TRANSACTION GO INSERT INTO test VALUES (1) GO INSERT INTO test VALUES ('aaa') GO COMMIT TRANSACTION GO SELECT * FROM test</pre>	<p>SQL Server returns an error when an attempted data type conversion fails. If the data type conversion error occurs inside a transaction, the transaction is terminated.</p> <p>Expect differences in behavior as compared to earlier versions of SQL Server if a data type conversion fails inside a transaction.</p>

## Installing SQL Server

### Utilities (Level 1)

SQL Server 6.x	SQL Server 2000
<p>The <b>probe</b> login, which required no password, was used by DB-Library and Windows NT Performance Monitor. The DB-Library two-phase commit library used the <b>probe</b> login to check on the status of distributed transactions. It was also used by Windows NT Performance Monitor to get statistics from SQL Server.</p>	<p>The <b>probe</b> login has been eliminated. Windows NT Performance Monitor will always use Windows NT Authentication, known earlier as integrated security, to connect to SQL Server. Ensure that your Windows NT username and password have the appropriate privileges to use Windows NT Performance Monitor.</p>

## Installing SQL Server

### Backup and Restore (Level 2)

SQL Server 6.x	SQL Server 2000
<p>Using the SKIP and INIT clauses of the DUMP statement together overwrote the contents of the backup device unconditionally.</p>	<p>The SKIP and INIT clauses of the BACKUP statement preserve the Microsoft Tape Format media header. In some situations, this prevents overwriting the backup contents. The FORMAT clause overwrites the media unconditionally, generating a new header, and is required for media used for the first time or when necessary to overwrite the media header.</p> <p>Expect different results as compared to earlier versions of SQL Server. If the media is empty, SKIP and INIT act the same as the FORMAT clause of the <a href="#">BACKUP</a> or DUMP statements and write a new media header. If the media is not empty, SKIP and INIT do not write a new media header.</p>
<p>The LOAD statement did not create the database automatically when restoring the database backup.</p>	<p>It is no longer necessary to create the database before restoring it. The RESTORE statement re-creates the database automatically, including all files. However, database devices are not re-created in <b>sysdevices</b>. These devices are supported only for backward compatibility. After restoration, databases originally created using devices (DISK INIT) appear as if they had been created using SQL Server 2000 file syntax.</p>

	<p>Expect different results as compared to earlier versions of SQL Server. Consider using the new syntax in <a href="#">CREATE DATABASE</a> and <a href="#">ALTER DATABASE</a> for specifying files.</p>
<p>The NO_LOG clause of DUMP was used only when you ran out of space in the database and could not use DUMP TRANSACTION WITH TRUNCATE_ONLY to purge the log. The NO_LOG clause removes the inactive part of the log without making a backup copy of it, and saves space by not logging the operation. The TRUNCATE_ONLY clause of the DUMP statement removed the inactive part of the log without making a backup copy of it.</p>	<p>The NO_LOG and TRUNCATE_ONLY clauses of RESTORE are synonyms. Both clauses of BACKUP now remove the inactive part of the log without making a backup copy of it and truncate the log.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect the NO_LOG and TRUNCATE_ONLY clauses of the <a href="#">BACKUP</a> or DUMP statements to behave identically.</p>
<p>Recovery of multiple transaction logs could be performed without special keywords in the LOAD statement.</p>	<p>It is no longer possible to restore multiple transaction logs without using the WITH clauses of the RESTORE statement.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use the appropriate <a href="#">RESTORE</a> syntax for restoring a database with multiple transaction logs as shown in the following examples. All but the last RESTORE statement should specify the NORECOVERY clause.</p>
<p>When loading a database, all database options of <b>sp_dboption</b> were unaffected and had to be set manually.</p>	<p>Changes to all <b>sp_dboption</b> database settings (except the <b>offline</b>, <b>merge</b>, <b>publish</b>, <b>published</b>, and <b>subscribed</b> settings) are logged, like any other</p>

change. When a database is restored and recovered, all database options of **sp\_dboption** are rolled forward. Every database option will be in its expected state at the time when recovery finished, consistent with the remainder of the database.

Expect different results as compared to earlier versions of SQL Server. It is no longer necessary to reset the database options after a RESTORE operation.

## Examples

### A. Restore a database by applying a full database backup and multiple transaction logs

This example restores a database with multiple transaction log backups.

```
RESTORE DATABASE mydb  
FROM mydb  
WITH NORECOVERY
```

```
RESTORE LOG mydb  
FROM mydb_log1  
WITH NORECOVERY
```

```
RESTORE LOG mydb  
FROM mydb_log2  
WITH RECOVERY
```

SQL Server 6.x	SQL Server 2000
A warm standby server could be brought up in read-only mode between recovery of each	A warm standby server can be brought up in read-only mode between transaction log restore operations if an

transaction log, provided that the **no chkpt. on recovery** option of **sp\_dboption** was enabled.

undo file is used.

Expect different results as compared to earlier versions of SQL Server. Use an undo file for a warm standby server using the STANDBY clause of [RESTORE](#), as shown in the following example.

## **B. Restore a database using the STANDBY clause and an undo file**

This example brings the server up to allow write operations on the databases by using a final, necessary RESTORE statement.

```
RESTORE DATABASE mydatabase
FROM mydb_backup
WITH NORECOVERY
```

```
RESTORE LOG mydb
FROM mydb_log1
WITH RECOVERY STANDBY (FILENAME = 'c:\mssql\data\mydbu
```

```
RESTORE LOG mydb
FROM mydb_log2
WITH RECOVERY STANDBY (FILENAME = 'c:\mssql\data\mydbu
```

```
RESTORE DATABASE mydb
WITH RECOVERY
```

## Installing SQL Server

### Bulk Copy (Level 2)

SQL Server 6.x	SQL Server 2000
<p>The <b>bcp</b> utility (using DB-Library) could import <b>datetime</b> or <b>smalldatetime</b> values in character-mode data files using:</p> <ul style="list-style-type: none"><li>• The default format used by DB-Library (mmm dd yyy hh:mmXX where XX is either A.M. or P.M.).</li><li>• Any format supported by <b>dbconvert</b> except the ODBC format.</li></ul> <p><b>bcp</b> exported character-mode data files with <b>datetime</b> and <b>smalldatetime</b> values by using the default DB-Library format.</p>	<p>The <b>bcp</b> utility (which uses ODBC) can import <b>datetime</b> and <b>smalldatetime</b> values in character-mode data files using:</p> <ul style="list-style-type: none"><li>• The default format used by DB-Library.</li><li>• The format used by ODBC (yyyy-mm-dd hh:mm:ss[.f...]).</li></ul> <p>However, <b>bcp</b> does not use other formats supported by <b>dbconvert</b>.</p> <p><b>bcp</b> exports <b>datetime</b> and <b>smalldatetime</b> values using the ODBC default format.</p> <p>Expect different results as compared to earlier versions of SQL Server. To bulk copy data in character mode between SQL Server 6.x and SQL Server 2000 servers, use the same <b>bcp</b> version (either SQL Server 6.x or SQL Server 2000) for both importing and exporting data.</p> <p>To export data from a SQL Server 2000 server into a character-mode data file and later import that data using a DB-Library bulk copy application, use the SQL Server 6.x version of <b>bcp</b>.</p> <p>For existing <b>datetime</b> or <b>smalldatetime</b> values in a character-mode data file in a</p>

	<p>format other than the DB-Library default:</p> <ul style="list-style-type: none"> <li>• Change the values to the DB-Library default format for continued use with SQL Server 6.x and SQL Server 2000 <b>bcp</b>.</li> <li>• Change the values to the ODBC format for use with SQL Server 2000 <b>bcp</b>.</li> </ul>
<p><b>bcp</b> exported <b>money</b> values in character mode data files using digit grouping symbols (for example, the comma in the United States when using the U.S. version of SQL Server, the US version of Microsoft Windows NT, and US settings) and two digits after the decimal point.</p>	<p><b>bcp</b> exports <b>money</b> values in character mode data files without digit grouping symbols, but with four digits after the decimal point.</p> <p>Expect different results as compared to earlier versions of SQL Server.</p> <p>To read character files created by version 6.x DB-Library <b>bcp</b> in SQL Server 2000, use the <b>-V</b> switch. For more information, see <a href="#">bcp Utility</a>.</p>

## Installing SQL Server

### Configuration Options (Level 2)

Administrative scripts may have used these configuration options. For more information about configuration options, see [sp\\_configure](#) and [Setting Configuration Options](#).

SQL Server 6.x	SQL Server 2000
<b>open objects</b> set the maximum number of database objects that can be open at one time on SQL Server.	Now an advanced option. Default value of 0 indicates automatic growth.  Expect different results as compared to earlier versions of SQL Server. Remove all references to <b>open objects</b> . For more information, see <a href="#">open objects Option</a> .
<b>user connections</b> set the maximum allowed number of simultaneous connections to SQL Server.	Now an advanced option. Default value of 0 indicates automatic growth.  Expect different results as compared to earlier versions of SQL Server. Remove all references to <a href="#">user connections Option</a> .

## Installing SQL Server

### Database Pages and Extents (Level 2)

SQL Server 6.x	SQL Server 2000
A database extent consisted of eight 2 KB pages.	<p>A database extent consists of eight 8 KB pages. Different objects can now share an extent or an object can have its own extent. A table and index both have a minimum of two pages.</p> <p>Expect different results as compared to earlier versions of SQL Server. Adjust disk space requirements for adequate database storage. For more information, see <a href="#">Pages and Extents</a>.</p>

## Installing SQL Server

### Data Types (Level 2)

SQL Server 6.x	SQL Server 2000
Conversion of <b>binary</b> or <b>varbinary</b> to <b>decimal</b> or <b>numeric</b> was explicit.	This conversion is implicit.  Expect different results as compared to earlier versions of SQL Server. Expect conversions of <b>binary</b> or <b>varbinary</b> to <b>decimal</b> or <b>numeric</b> to be implicit. For more information about data type conversions, see <a href="#">CAST and CONVERT</a> . For more information about system-supplied data types, see <a href="#">Data Types</a> .
Conversion of <b>binary</b> or <b>varbinary</b> to <b>smallmoney</b> was not allowed.	This conversion is allowed.  Expect different results as compared to earlier versions of SQL Server. Use as appropriate.
Conversion of <b>datetime</b> or <b>smalldatetime</b> to <b>decimal</b> , <b>numeric</b> , <b>float</b> , <b>real</b> , <b>int</b> , <b>smallint</b> , <b>tinyint</b> , <b>money</b> , <b>smallmoney</b> , or <b>bit</b> was not allowed.	This conversion is allowed.  Expect different results as compared to earlier versions of SQL Server. Use as appropriate.
Conversion of <b>float</b> or <b>real</b> to <b>binary</b> or <b>varbinary</b> was not allowed.	This conversion is allowed.  Expect different results as compared to earlier versions of SQL Server. Use as appropriate.
Conversion of <b>money</b> or <b>smallmoney</b> to <b>char</b> or <b>varchar</b> was implicit.	This conversion is explicit.  Expect different results as compared to earlier versions of SQL Server. Expect conversions of <b>money</b> or <b>smallmoney</b> to <b>char</b> or <b>varchar</b> to be explicit.

<p>Conversion of <b>bit</b> to <b>money</b> or <b>smallmoney</b> was not allowed.</p>	<p>This conversion is allowed.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use as appropriate.</p>
<p><b>sysname</b> data type was <b>varchar(30)</b>.</p>	<p><b>sysname</b> data type is <b>nvarchar(128)</b>, which allows for 128 Unicode characters.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect any columns or local variables defined as <b>sysname</b> to allow Unicode data. For more information about <b>sysname</b>, see <a href="#">Data Types</a>.</p>
<p>No direct support for the <b>nchar</b>, <b>nvarchar</b>, and <b>ntext</b> Unicode data types because maximum storage was 255 bytes for <b>char</b>, <b>binary</b>, <b>varchar</b>, and <b>varbinary</b> data types.</p>	<p>The maximum number of bytes that can be stored in <b>char</b>, <b>binary</b>, <b>varchar</b>, and <b>varbinary</b> data types is increased to 8,000. SQL Server 2000 clients fully support the <b>nchar</b>, <b>nvarchar</b>, and <b>ntext</b> data types. SQL Server 6.x clients accessing SQL Server 2000 with these Unicode data types will experience these results:</p> <p><b>nvarchar</b> data is returned as <b>varchar</b> and <b>nchar</b> data is returned as <b>char</b>. <b>nvarchar</b> and <b>nchar</b> values longer than 255 double-byte characters are truncated to 255 single-byte characters.</p> <p>Attempting to access <b>ntext</b> data causes SQL Server to issue a 4004 error. <b>ntext</b> data cannot be sent to version 6.x clients.</p> <p><b>char</b>, <b>varchar</b>, <b>binary</b>, and <b>varbinary</b> values longer than 255 bytes are truncated to 255 bytes.</p>

	<p>Expect different results as compared to earlier versions of SQL Server. Expect changes in data when accessing SQL Server 2000 data from version 6.x clients. To eliminate these differences, upgrade the clients to SQL Server 2000 client software.</p>
<p>Using CONVERT to convert an empty string to <b>int</b> (CONVERT(int, "")) or <b>float</b> (CONVERT(float, "")) returned a zero.</p>	<p>Using CAST or CONVERT to convert an empty string to <b>int</b> (CAST("" AS int)) or <b>float</b> (CAST("" AS float)) returns an error message.</p> <p>Expect different results as compared to earlier versions of SQL Server.</p>

## Installing SQL Server

### DB-Library (Level 2)

SQL Server 6.x	SQL Server 2000
<p>When connected to a version 6.x SQL Server, a call to <b>dbcursorfetchex</b> resulting in a cursor position after the end of the cursor result set returned:</p> <ul style="list-style-type: none"><li>• FAIL, with either a keyset or an insensitive cursor.</li><li>• SUCCEED, with all row status indicators set to 0, with either a dynamic or forward cursor.</li></ul>	<p>A call to <b>dbcursorfetchex</b> resulting in a cursor position after the end of the cursor result set returns SUCCEED. All row status indicators are set to 0. This behavior applies to all types of cursors.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect different results, compared to SQL Server 6.x, when a call to <a href="#">dbcursorfetchex</a> results in a cursor position after the end of the cursor result set. To achieve SQL Server 6.x behavior, use compatibility level 65.</p>

## Installing SQL Server

### Empty Strings (Level 2)

SQL Server 6.x	SQL Server 2000
<p>An empty string could be interpreted as either a NULL or a single blank character.</p>	<p>Interpretation of an empty string is controlled by the compatibility level, which is set with the <b>sp_dbcmplevel</b> system stored procedure. If the compatibility level is 65 or lower, SQL Server interprets empty strings as single spaces. If the compatibility level is 70 or 80, SQL Server interprets empty strings as empty strings. For more information, see <a href="#">sp_dbcmplevel</a>.</p> <p>Expect differences in interpretation of empty strings compared to earlier versions of SQL Server. Transact-SQL functions and statements affected by the setting of <b>sp_dbcmplevel</b> include <a href="#">CHARINDEX</a>, <a href="#">DATALENGTH</a>, <a href="#">LEFT</a>, <a href="#">LTRIM</a>, <a href="#">PATINDEX</a>, <a href="#">REPLICATE</a>, <a href="#">RIGHT</a>, <a href="#">RTRIM</a>, <a href="#">SPACE</a>, <a href="#">SUBSTRING</a>, and <a href="#">UPDATETEXT</a>.</p>

## Installing SQL Server

### Indexes (Level 2)

SQL Server 6.x	SQL Server 2000
<p>The DROP INDEX statement dropped the pages holding the clustered index B-tree when used on a clustered index.</p>	<p>On a clustered index, the DROP INDEX statement must rebuild all nonclustered indexes. SQL Server must also replace the clustered index keys in the nonclustered leaf rows with row pointers.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use the DROP_EXISTING clause of the CREATE INDEX statement if, for example, dropping or re-creating clustered indexes to set a new fill factor value.</p>

## Installing SQL Server

### INSERT (Level 2)

SQL Server 6.x	SQL Server 2000
<p>An INSERT x SELECT INTO Y statement ignored table Y and inserted the SELECT results into table X, as shown.</p> <pre>INSERT X SELECT select_list INTO Y</pre>	<p>The INSERT...SELECT INTO syntax is retained only when the compatibility setting is equal to 60 or 65. If the compatibility setting is 70 or 80 and a similar query is executed, SQL Server returns a syntax error.</p> <p>Expect different results as compared to earlier versions of SQL Server. Set the compatibility setting to 60 or 65 by executing <a href="#">sp_dbcmptlevel</a> to retain version 6.x functionality for queries using INSERT ...SELECT INTO syntax. Otherwise, use a compatibility setting of 80.</p>
<p>In an INSERT statement, a SELECT statement returning a scalar value was allowed in the VALUES clause.</p>	<p>The INSERT statement cannot have a SELECT statement in the VALUES clause as one of the values to be inserted. The version 6.x supportability is available only when the compatibility setting is equal to 60 or 65.</p> <p>Expect different results as compared to earlier versions of SQL Server. Set the compatibility setting to 60 or 65 by executing <a href="#">sp_dbcmptlevel</a> to retain version 6.x functionality for using a SELECT statement in the VALUES clause of an <a href="#">INSERT</a> statement. Otherwise, use a compatibility setting of 80.</p>

A ROLLBACK statement in a stored procedure referenced in an INSERT *table* EXEC *procedure* statement caused the INSERT to be rolled back, but the batch continued.

A ROLLBACK statement in the stored procedure referenced by an INSERT...EXEC statement causes the entire transaction to be rolled back and the batch stops executing. The version 6.x supportability is available only when the compatibility setting is equal to 60 or 65.

Expect different results as compared to earlier versions of SQL Server. Set the compatibility setting to 60 or 65 by executing [sp\\_dbcmptlevel](#) to retain version 6.x functionality for ROLLBACK statement behavior inside an [INSERT...EXEC](#) statement. Otherwise, use a compatibility setting of 80.

## Installing SQL Server

### Keyset Cursors (Level 2)

SQL Server 6.x	SQL Server 2000
<p>When using a keyset cursor, a row deletion followed by a row insertion using the same key as the deleted row caused the inserted row to occupy the slot of the original row.</p>	<p>When using a keyset cursor, a row deletion followed by a row insertion with the same key as the deleted row allows the original row to remain empty and the newly inserted row to be inserted at the end.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect a change in behavior when inserting and deleting rows with the same key values when using keyset cursors.</p>

## Installing SQL Server

### **LTRIM and RTRIM Trimming Functions (Level 2)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
<p>The LTRIM and RTRIM functions returned NULL in queries using zero-length strings:</p> <pre>SELECT RTRIM("")</pre> <pre>SELECT DATALENGTH(RTRIM(""))</pre>	<p>Zero-length strings are supported. The queries shown return nonnull values; the first returns "" and the second returns 0.</p> <p>Expect different results as compared to earlier versions of SQL Server. <a href="#">LTRIM</a> and <a href="#">RTRIM</a> provide different output from earlier versions of SQL Server.</p>

## Installing SQL Server

### ODBC (Level 2)

SQL Server 6.x	SQL Server 2000
<p data-bbox="191 436 797 659">SQL_ERROR was returned by <b>SQLExecute</b>, <b>SQLExecDirect</b>, or <b>SQLParamData</b> when extended stored procedures or batches met the following criteria:</p> <ul data-bbox="289 688 797 1213" style="list-style-type: none"><li data-bbox="289 688 797 961">• The first data-returning statement caused an error (either by a run-time error or a RAISERROR statement with severity greater than or equal to 11).</li><li data-bbox="289 1037 797 1213">• There was data from any other statement, even a simple RETURN statement, after the error-causing statement.</li></ul> <p data-bbox="191 1247 797 1373">Due to the SQL_ERROR return code, the statement handle was available for use immediately.</p>	<p data-bbox="797 436 1354 751">SQL_SUCCESS_WITH_INFO is returned when an ODBC 3.x application uses the ODBC SQL Server 3.51-compliant driver included with this release (using <b>SQLExecute</b>, <b>SQLExecDirect</b>, or <b>SQLParamData</b>).</p> <p data-bbox="797 785 1354 995">Due to the SQL_SUCCESS_WITH_INFO return code, process the results for that statement handle before it is available for use.</p> <p data-bbox="797 1037 1354 1394">Expect different results as compared to earlier versions of SQL Server. Handle SQL_SUCCESS_WITH_INFO using <b>SQLGetDiagRec</b>, and then call <b>SQLMoreResults</b> to process the remaining results, as appropriate.</p>

## Installing SQL Server

### **RIGHT (Level 2)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
RIGHT was a reserved word.	<p>RIGHT is a reserved keyword and should not be used for database object names (unless using identifiers). For more information about SQL Server 2000 reserved keywords, see <a href="#">Using Reserved Keywords</a>.</p> <p>Expect different results as compared to earlier versions of SQL Server because RIGHT is now a reserved keyword. For more information about using RIGHT with identifiers, see <a href="#">Using Identifiers</a>.</p>

## Installing SQL Server

### Security (Level 2)

SQL Server 6.x	SQL Server 2000
<p>The GRANT and REVOKE statements granted and revoked permissions, respectively. The REVOKE statement denied a permission to a single user that was granted to the user's group.</p>	<p>The security model uses DENY in addition to GRANT and REVOKE. REVOKE has changed to remove a previously granted or denied permission. DENY creates an entry in the security system that denies a permission from a security account and prevents the user, group, or role from inheriting the permission through its group and role memberships. The REVOKE statement can no longer be used to deny permission to a user whose group has permission. Use the DENY statement to deny permissions explicitly to a specific user or group.</p> <p>Expect different results as compared to earlier versions of SQL Server. Recognize that scripts using the SQL Server 6.x security model using GRANT and REVOKE behave differently than scripts using the current model of <a href="#">GRANT</a>, <a href="#">REVOKE</a>, and <a href="#">DENY</a> if REVOKE was used to deny permissions to selected members of a group.</p>
<p>When executing an RPC, logins using integrated security mode referred to an internal login name with the backslashes (\) translated to underscores (_). For example, \Domain\Joe was translated to</p>	<p>Those servers upgraded to SQL Server that execute RPC calls no longer translate backslashes to underscores when using Windows NT Authentication. To use the SQL Server version 6.x naming convention for login</p>

Domain\_Joe.

names, use **sp\_addlinkedsrvlogin** to map the backslash version of the username to an underscore version.

Expect different results as compared to earlier versions of SQL Server. Add references for [sp\\_addlinkedsrvlogin](#) to translate backslash version login names to underscore versions to maintain version 6.x login translations when the sending server of an RPC uses SQL Server 2000.

## Examples

### A. Map specific backslash login to underscore login

This example maps the \LONDON1\nancyd login name to LONDON1\_nancyd:

```
sp_addlinkedsrvlogin 'receiving_server_name',  
    false,  
    'LONDON1\nancyd',  
    'LONDON1_nancyd', NULL
```

### B. Map specific backslash login to sa login

This example maps Nancy's LONDON1 login to the **sa** login, because Nancy's domain login is part of the built-in administrators group:

```
sp_addlinkedsrvlogin 'receiving_server_name',  
    false,  
    'LONDON1\nancyd',  
    'sa', NULL
```

## Installing SQL Server

### **SELECT (Level 2)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
A SELECT statement without an ORDER BY clause returned the rows in an apparent ordered set.	<p>An explicit ORDER BY clause for a SELECT statement is required to ensure any useful ordering of data. In addition, the exact results depend upon the collation being used.</p> <p>Expect different results as compared to earlier versions of SQL Server. Add an explicit ORDER BY clause to all <a href="#">SELECT</a> statements needing to produce ordered rows.</p>

## Installing SQL Server

### SET SHOWPLAN (Level 2)

SQL Server 6.x	SQL Server 2000
<p>When SET SHOWPLAN was set ON, SQL Server executed Transact-SQL statements.</p>	<p>When set ON, the SET SHOWPLAN_ALL and SET SHOWPLAN_TEXT statements, which replace SET SHOWPLAN, do not execute Transact-SQL statements.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect a difference in behavior when <a href="#">SET SHOWPLAN_ALL</a> or <a href="#">SET SHOWPLAN_TEXT</a> are set to ON.</p>

## Installing SQL Server

### System Tables (Level 2)

SQL Server 6.x	SQL Server 2000
System tables were used internally by SQL Server for a wide range of uses.	Some system tables have had minor changes, while others have been replaced by Information Schema Views that provide the same information.  Expect different results as compared to earlier versions of SQL Server. Use the provided <a href="#">Information Schema Views</a> or ODBC catalog <a href="#">system stored procedures</a> to obtain system catalog information.
The <b>logptr</b> column of <b>sysdatabases</b> was a pointer to the transaction log.	The <b>logptr</b> column has been renamed to <b>status2</b> . Remove all references of the <b>logptr</b> column of <a href="#">sysdatabases</a> and replace with references to the <b>status2</b> column.
The <b>dumptrdate</b> column of <b>sysdatabases</b> was the date of the last DUMP TRANSACTION.	This column is now <b>Reserved</b> .  Expect different results as compared to earlier versions of SQL Server. Remove all references to the <b>dumptrdate</b> column of <a href="#">sysdatabases</a> .
The <b>langid</b> column of <b>sysmessages</b> contained the SQL Server message group ID.	The <b>langid</b> column has been renamed to <b>msglangid</b> .  Expect different results as compared to earlier versions of SQL Server. Remove all references of the <b>langid</b> column of <a href="#">sysmessages</a> and replace with references to the <b>msglangid</b> column.
A NULL value for the <b>language</b> column of the <b>syslogins</b> table was	A NULL value for the <b>language</b> column is no longer equivalent to <b>us_english</b> .

<p>equivalent to specifying <b>us_english</b>.</p>	<p>Expect different results as compared to earlier versions of SQL Server. Remove all NULL values for the <b>language</b> column of <a href="#">syslogins</a> and replace with the name of the language to be used.</p>
<p>System tables obtained their column values by insertion of a specific value (SQL Server 2000 uses computed columns in many system and user-defined tables.)</p>	<p>System tables (and user-defined tables) can now use computed columns.</p> <p>Expect different results as compared to earlier versions of SQL Server. SQL Server version 6.5 queries involving table hints and system tables may still produce the same result set, but may behave differently in SQL Server 2000. For example, the query may still wait for some locks even if the NOLOCK table hint has been specified in the query's FROM clause.</p>

## Installing SQL Server

### Table Hints (Level 2)

SQL Server 6.x	SQL Server 2000
<p>These table hints (previously called optimizer hints) could be specified as just the keyword following the FROM clause:</p> <p>FASTFIRSTROW, HOLDLOCK, INDEX, NOLOCK, PAGLOCK, TABLOCK, TABLOCKX, and UPDLOCK.</p>	<p>Table hints must be specified following the FROM clause using a WITH clause. Table hints must be enclosed in parentheses.</p> <p>Expect different results as compared to earlier versions of SQL Server. For more information, see <a href="#">DELETE</a>, <a href="#">FROM</a>, <a href="#">INSERT</a>, <a href="#">SELECT</a>, and <a href="#">UPDATE</a>.</p>

## Installing SQL Server

### Transactions (Level 2)

SQL Server 6.x	SQL Server 2000
<p>When <code>CURSOR_CLOSE_ON_COMMIT</code> was set OFF, a ROLLBACK statement did not close a Transact-SQL cursor defined with the <code>DECLARE CURSOR</code> statement. Server cursors opened through database API functions were also left open after a ROLLBACK statement.</p>	<p>When <code>CURSOR_CLOSE_ON_COMMIT</code> is set OFF, a ROLLBACK statement closes any Transact-SQL cursor defined with the SQL-92 form of the <code>DECLARE CURSOR</code> statement, unless the <code>DECLARE CURSOR</code> statement contains either the <code>INSENSITIVE</code> or <code>STATIC</code> keywords. All API server cursors are also closed unless they have been defined as <code>STATIC</code> cursors (such as using the ODBC <code>SQL_CURSOR_STATIC</code> attribute).</p> <p>Expect different results as compared to earlier versions of SQL Server. Reopen all cursors after issuing a ROLLBACK statement.</p>
<p>The <code>REPEATABLE READ</code> clause of the <code>SET TRANSACTION ISOLATION LEVEL</code> statement behaved identically to the <code>SERIALIZABLE</code> clause. There was no way to ensure repeatable reads without also protecting against phantoms (after a rollback, the value read logically never existed). Transactions that required <code>REPEATABLE READ</code> semantics had to pay the additional concurrency penalty of serializability.</p>	<p>The <code>REPEATABLE READ</code> clause now does not necessarily protect against phantoms. Serializable transactions, set using the <code>SERIALIZABLE</code> clause of <code>SET TRANSACTION ISOLATION LEVEL</code>, allow less concurrency than the <code>REPEATABLE READ</code> clause because they protect against phantoms.</p> <p>Expect different results as compared to earlier versions of SQL Server. Many applications only need <code>REPEATABLE READ</code> semantics for correct operation. Use the <code>REPEATABLE READ</code> clause</p>

of [SET TRANSACTION ISOLATION LEVEL](#) for applications requiring REPEATABLE READ semantics but that do not need phantom protection. If phantom protection is required, use the SERIALIZABLE clause.

Here is a summary of phantom protection for both SQL Server versions 6.5 and SQL Server 2000 using SET TRANSACTION ISOLATION LEVEL.

<b>Phantom protection</b>	<b>SQL Server 6.5</b>	<b>SQL Server 2000</b>
REPEATABLE READ	Yes	No
SERIALIZABLE	Yes	Yes

## Triggers and System Stored Procedures (Level 2)

SQL Server 6.x	SQL Server 2000
<p><b>sp_helpsql</b> provided syntax for Transact-SQL statements, system stored procedures, and other special topics.</p>	<p><b>sp_helpsql</b> is included, but no longer returns syntax information for Transact-SQL statements or system stored procedures. Executing <b>sp_helpsql</b> produces a message that recommends obtaining syntax information from Online Help.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use SQL Server Books Online for the syntax of Transact-SQL statements and system stored procedures.</p>
<p>Only one trigger for each data modification event (INSERT, UPDATE, or DELETE) was allowed for each table. If a new trigger was created for a specific data modification event, it replaced the previous trigger.</p>	<p>Microsoft® SQL Server™ allows multiple triggers to be created for each data modification event (DELETE, INSERT, or UPDATE). For example, if CREATE TRIGGER FOR UPDATE is executed for a table that already has an UPDATE trigger, an additional UPDATE trigger is created.</p> <p>Expect different results as compared to earlier versions of SQL Server. Enable multiple triggers by setting the compatibility level to 80 in <b>sp_dbcmptlevel</b>. Retain SQL Server 6.x behavior by setting the compatibility level to 60 or 65. For more information, see <a href="#">sp_dbcmptlevel</a> and <a href="#">CREATE TRIGGER</a>.</p>

<p>If a trigger modified the table on which it was defined, the triggers were not invoked recursively for that modification.</p>	<p>SQL Server allows recursive invocation of triggers.</p> <p>Expect different results as compared to earlier versions of SQL Server. Enable recursive triggers by setting the <code>RECURSIVE_TRIGGERS</code> database option. For more information about recursive and nested triggers, see <a href="#">Nested Triggers</a>.</p>
<p>Several parameters of <code>sp_create_removable</code> referred to devices.</p>	<p>Devices have been replaced with files and filegroups.</p> <p>Expect different results as compared to earlier versions of SQL Server. Replace all device references in <a href="#">sp_create_removable</a> with references to filegroups</p>
<p>References to <b>text</b> or <b>image</b> columns in either the <b>inserted</b> or <b>deleted</b> tables appeared as NULL.</p>	<p>References to <b>text</b> or <b>image</b> columns in both the <b>inserted</b> and <b>deleted</b> tables are no longer allowed unless the compatibility level setting of <code>sp_dbcmplevel</code> is 60 or 65.</p> <p>Expect a difference in behavior when referring to <b>text</b> or <b>image</b> columns in <b>inserted</b> and <b>deleted</b> tables when using <a href="#">CREATE TRIGGER</a>, depending on the setting of <a href="#">sp_dbcmplevel</a>.</p>
<p>SQL Server searched the current database followed by a search in <b>master</b> for a stored procedure using the <code>sp_</code> prefix.</p>	<p>Stored procedures with the prefix <code>sp_</code> are first looked up in <b>master</b>. If a user-defined stored procedure has the same name as a system-supplied stored procedure residing in <b>master</b>, SQL Server always finds the system-supplied stored procedure.</p>

	<p>Expect different results as compared to earlier versions of SQL Server. Expect a difference in behavior when calling user-defined stored procedures with the <b>sp_</b> prefix. Either explicitly qualify the name of the user-defined stored procedure, or rename the user-defined stored procedure.</p>
<p>The settings of SET ANSI_NULLS and SET QUOTED_IDENTIFIER statements were active only during the session that changed either option.</p>	<p>The settings of both SET QUOTED_IDENTIFIER and SET ANSI_NULLS are saved when a stored procedure is created or altered. These original settings are enabled when the stored procedure is executed, and any client session settings are restored afterward. Within the stored procedure, any changes to SET ANSI_NULLS do not take effect until after the stored procedure executes.</p> <p>Expect different results as compared to earlier versions of SQL Server. Develop databases or applications with one setting for <a href="#">SET QUOTED_IDENTIFIER</a>, <a href="#">SET ANSI_NULLS</a>, and all other pertinent SET options. If a client session changes SET options, do so outside of stored procedures.</p>
<p>When executing remote stored procedures, these procedures may have assumed non-standard behavior for the options set by SET ANSI_DEFAULTS. In addition, remote stored procedures may not have explicitly set these</p>	<p>When executing remote stored procedures, these procedures are executed with SET ANSI_DEFAULTS set to ON.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect a difference in behavior when executing</p>

options (set by SET ANSI\_DEFAULTS) to OFF.

remote stored procedures if non-standard settings were used with [SET ANSI\\_DEFAULTS](#), or if options were not explicitly set to OFF.

## Installing SQL Server

### UPDATE (Level 2)

SQL Server 6.x	SQL Server 2000
<p>When ARITHABORT was set to OFF, an UPDATE statement encountering an arithmetic overflow condition would set the updated value to NULL, or skip the update if the value belonged to a nonnull column.</p>	<p>When SET ARITHABORT is OFF and an INSERT, UPDATE, or DELETE statement encounters an arithmetic error, SQL Server inserts or updates a NULL value. If the target column is not nullable, the insert or update action fails and the user receives an error.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use the <a href="#">@@ERROR</a> function to test for errors after <a href="#">UPDATE</a> or <a href="#">INSERT</a> statements.</p>

## Installing SQL Server

### UPDATETEXT (Level 2)

SQL Server 6.x	SQL Server 2000
UPDATETEXT initialized <b>text</b> columns to NULL, allocating a full 2K page.	<p>If the compatibility level setting of <b>sp_dbcmplevel</b> is 65, UPDATETEXT initializes <b>text</b> columns to NULL. However, if the compatibility level setting is 70 or 80, WRITETEXT initializes <b>text</b> columns to NULL; UPDATETEXT initializes <b>text</b> columns to an empty string.</p> <p>Expect differences in behavior when initializing <b>text</b> values to NULL (using <a href="#">UPDATETEXT</a> or <a href="#">WRITETEXT</a>) depending on the compatibility level setting of <a href="#">sp_dbcmplevel</a>.</p>

## Installing SQL Server

### Views (Level 2)

SQL Server 6.x	SQL Server 2000
Updatable views were restricted to modifications that affected only one table.	<p>Updatable views can modify more than one table involved in the view. The DELETE, INSERT, and UPDATE statements can reference a view as long as SQL Server can translate the user's update request unambiguously to updates in the base tables referenced in the view's definition.</p> <p>Expect differences in behavior when working with updatable views with more than one table involved in the <a href="#">DELETE</a>, <a href="#">INSERT</a>, or <a href="#">UPDATE</a> statements.</p>

## Installing SQL Server

### Backup and Restore (Level 3)

SQL Server 6.x	SQL Server 2000
<p>The DUMP statement created database or transaction log backups (dumps).</p>	<p>The DUMP DATABASE and DUMP TRANSACTION statements are synonymous with BACKUP DATABASE and BACKUP LOG statements. Support for the DUMP DATABASE and DUMP TRANSACTION statements may be removed in a future release.</p> <p>Consider removing all references of DUMP DATABASE and replacing with references to <a href="#">BACKUP DATABASE</a>. Consider removing all references of DUMP TRANSACTION and replacing with references to <a href="#">BACKUP LOG</a>.</p>
<p>The LOAD statement restored or loaded database or transaction log backups (dumps).</p>	<p>The LOAD DATABASE and LOAD TRANSACTION statements are synonymous with the RESTORE DATABASE and RESTORE LOG statements. Support for the LOAD DATABASE and LOAD TRANSACTION statements may be removed in a future release.</p> <p>Consider removing all references of LOAD DATABASE and replacing with references to RESTORE DATABASE. Consider removing all references of LOAD TRANSACTION and replacing with references to RESTORE LOG. For more information about RESTORE DATABASE, see <a href="#">RESTORE</a>.</p>

The CREATE DATABASE ...FOR LOAD statement syntax created a destination database before its restoration from a database backup and prevented anyone from using the database between the CREATE DATABASE, ALTER DATABASE, and LOAD statements.

The CREATE DATABASE ...FOR LOAD syntax is supported for backward compatibility only. However, because SQL Server now creates the destination database within a restore operation, it is recommended that the destination database not be created before executing the restore operation.

Do not create the database prior to restoring it.

## Installing SQL Server

### Database Options (Level 3)

SQL Server 6.x	SQL Server 2000
The <b>publish</b> option of <b>sp_dboption</b> enabled or disabled publishing in a database.	<b>sp_replicationdboption</b> should be used to enable or disable publishing in a database.  Remove all references of <b>sp_dboption publish</b> and replace with references to <a href="#"><b>sp_replicationdboption</b></a> .

## Installing SQL Server

### DBCC (Level 3)

SQL Server 6.x	SQL Server 2000
DBCC NEWALLOC checked data and index pages against corresponding extent structures.	DBCC NEWALLOC is supported for backward compatibility only and is identical to DBCC CHECKALLOC.  Consider removing all references of DBCC NEWALLOC and replacing with references to <a href="#">DBCC CHECKALLOC</a> .
DBCC ROWLOCK dynamically enabled Insert Row Locking (IRL) operation on tables.	Row-level locking is automatic. DBCC ROWLOCK available for backward compatibility only.  Consider removing all references of DBCC ROWLOCK.
DBCC TEXTALL selected tables in the database that had <b>text</b> or <b>image</b> columns and ran DBCC TEXTALLOC on them.	DBCC CHECKDB checks the consistency of <b>text</b> , <b>ntext</b> , and <b>image</b> columns in a database. DBCC TEXTALL is available for backward compatibility only.  Consider removing all references of DBCC TEXTALL and replacing with references to <a href="#">DBCC CHECKDB</a> .
DBCC TEXTALLOC checked the allocation of <b>text</b> or <b>image</b> columns for a table.	DBCC CHECKTABLE checks the integrity of the data, index, <b>text</b> , <b>ntext</b> , and <b>image</b> pages for the specified table. DBCC TEXTALLOC is available for backward compatibility only.  Consider removing all references of DBCC TEXTALLOC and replacing

	with references to <a href="#">DBCC CHECKTABLE</a> .
DBCC DBREPAIR dropped the specified, and usually damaged, database.	Use DROP DATABASE to drop or remove a SQL Server database. DBCC DBREPAIR is available for backward compatibility only.  Consider removing all references of DBCC DBREPAIR and replacing with references to <a href="#">DROP DATABASE</a> .

## Installing SQL Server

### Devices (Level 3)

The database architecture of Microsoft® SQL Server™ 2000 differs from the database architecture of SQL Server 6.x. In SQL Server 2000:

- Operating system files replace database devices.
- Data files and transaction logs cannot co-exist on the same operating system file.
- A single operating system file cannot be shared by multiple databases.

For more information about database architecture, see [Overview of SQL Server Architecture](#).

SQL Server 6.x	SQL Server 2000
DISK INIT created database or transaction log devices. When DISK INIT followed either a CREATE DATABASE or ALTER DATABASE statement, SQL Server used the specified devices for storing the specified database or transaction log.	<p>The CREATE DATABASE statement syntax and ALTER DATABASE statement syntax both allow the creation of separate data and log files. Both CREATE DATABASE and ALTER DATABASE create operating system files and databases in a single step (generating a log file automatically, if none is specified with the LOG ON clause).</p> <p>Consider removing all references to DISK INIT and replacing with references to either <a href="#">CREATE DATABASE</a> or <a href="#">ALTER DATABASE</a>.</p> <p>DISK INIT has limited support in SQL Server 2000. Existing scripts will run as long as they do not have data and log sharing of the same data files.</p>

DISK REINIT restored device entries to appropriate system tables when the device entry was missing from <b>sysdevices</b> .	Removed; no replacement.  Consider removing all references to DISK REINIT.
<b>sp_logdevice</b> put <b>syslogs</b> (contains the transaction log) on a separate database device. To add another log segment to a database with an existing log segment, it was necessary to execute DISK INIT followed by <b>sp_logdevice</b> .	Removed. The CREATE DATABASE statement creates a log file on a new operating system file.  Consider removing all references to <b>sp_logdevice</b> and replacing with references to <a href="#">CREATE DATABASE</a> . SQL Server 6.x scripts using the LOG ON clause of CREATE DATABASE will work as expected. Scripts without the LOG ON clause of CREATE DATABASE will have a log file generated automatically.
Devices created using DISK INIT and CREATE DATABASE could be dropped only by using <b>sp_dropdevice</b> .	Databases created without DISK INIT before CREATE DATABASE can be dropped with DROP DATABASE; otherwise, use <b>sp_dropdevice</b> .  Use <b>sp_dropdevice</b> when using DISK INIT, followed by <a href="#">CREATE DATABASE</a> .

## Examples

### A. Use both DISK INIT and CREATE DATABASE syntax

This example uses DISK INIT and CREATE DATABASE and works in SQL Server version 6.5 and SQL Server 2000:

```
DISK INIT name = 'testdb_data',
  physname = 'c:\testdb_data.dat',
  vdevno = 9,
  size = 10240
```

```
DISK INIT name = 'testdb_log',  
  physname = 'c:\testdb_log.dat',  
  vdevno = 8,  
  size = 10240  
CREATE DATABASE testdb  
  ON testdb_data = 10  
  LOG ON testdb_log = 10  
GO
```

## **B. Use of `sp_logdevice` and `CREATE DATABASE` in SQL Server 2000 fails**

In earlier versions of SQL Server, this script created a 20 MB database consisting of the two files named **testdb\_data** and **testdb\_log**. This script also moved the transaction log to the **testdb\_log** device by using **sp\_logdevice**.

**Note** Scripts like this one were usually generated by the SQL Server 6.5 **sp\_help\_revdatabase** system stored procedure, which used **sp\_logdevice** to ensure the proper device layout for database restores. Because SQL Server 2000 creates the database when it is restored, scripts such as these are no longer necessary.

-- SQL Server 6.x example.

```
DISK INIT name = 'testdb_data',  
  physname = 'c:\testdb_data.dat',  
  vdevno = 9,  
  size = 10240  
DISK INIT name = 'testdb_log',  
  physname = 'c:\testdb_log.dat',  
  vdevno = 8,  
  size = 10240  
CREATE DATABASE testdb on testdb_data = 10, testdb_log = 10  
-- Use sp_logdevice to move the log to the testdb_log device.  
EXEC sp_logdevice testdb, testdb_log
```

In SQL Server 2000, the above script does not work the same as in SQL Server

6.x because **sp\_logdevice** no longer exists.

In SQL Server 2000, this script creates a 20 MB database consisting of the two files named **testdb\_data** and **testdb\_log**. In addition, SQL Server generates a log file automatically, which is 25 percent of the database size. In the following script (using the devices created earlier), a 10 MB log file is generated automatically:

```
CREATE DATABASE testdb on testdb_data = 10, testdb_log = 10
```

### **C. Use CREATE DATABASE syntax only**

Using the SQL Server 2000 CREATE DATABASE syntax, the database from the earlier example could be created as follows:

```
CREATE DATABASE testdb ON (name = 'testdb_data',  
    filename = 'd:\testdb_data.dat', size = 10)  
    LOG ON (name = 'testdb_log', filename = 'd:\testdb_log.dat',  
    size = 10)
```

## Installing SQL Server

### Open Data Services (Level 3)

The Open Data Services gateway functions, macros, and events listed in the table are no longer supported.

<b>Function/macro name</b>
srv_ackattention
srv_config
srv_config_alloc
srv_errhandle
srv_event
srv_eventdata
srv_getconfig
srv_handle
srv_init
srv_langcpy
srv_langlen
srv_langptr
srv_log
srv_post_handle
srv_pre_handle
srv_run
srv_setevent
srv_terminatethread

## Installing SQL Server

### Query Performance (Level 3)

SQL Server 6.x	SQL Server 2000
Queries could include a server user ID (SUID) without performance implications, as shown in the following table.	Queries using SUIDs continue to run and produce the same results as in earlier versions of SQL Server. However, there is a severe performance penalty because SUIDs are no longer native to the new security design.  Consider removing all references to SUIDs and replacing with references to security identification numbers (SIDs) (as shown in the following table) to avoid degradation in query performance.

SQL Server 6.x SUID	Replace with SQL Server 2000 SID
SUSER_ID	<a href="#">SUSER_SID</a> , which returns a SID
SUSER_NAME	<a href="#">SUSER_SNAME</a> , which accepts a SID as input
syslogins.suid	<a href="#">syslogins.sid</a>
sysdatabases.suid	<a href="#">sysdatabases.sid</a>
sysremotelogins.suid	<a href="#">sysremotelogins.sid</a>
sysusers.suid	<a href="#">sysusers.sid</a>
sysalternates.suid	<a href="#">sysusers.isaliased</a>
sysalternates.altsuid	<a href="#">sysusers.isaliased</a>

### Examples

#### A. Use SIDs and SUIDs to display login names of users in sysusers

This example shows SQL Server 6.x queries that displayed the login names of all users in **sysusers**:

```
SELECT L.name
FROM master.dbo.syslogins L, sysusers U
WHERE L.suid = U.suid
-- Or
SELECT suser_name(suid) AS name
FROM sysusers
```

Here are the queries rewritten to use SIDs rather than SUIDs:

```
SELECT L.loginname
FROM master.dbo.syslogins L, sysusers U
WHERE L.sid = U.sid
-- Or
SELECT suser_sname(sid) AS name
FROM sysusers
```

## Installing SQL Server

### Security (Level 3)

SQL Server 6.x	SQL Server 2000
<p>The ON {<i>table</i>   <i>view</i>} (<i>column</i> [...<i>n</i>]) syntax for the GRANT statement assigned the specified permissions to the columns given for the specified table or view.</p>	<p>The ON {<i>table</i>   <i>view</i>} (<i>column</i> [...<i>n</i>]) syntax for the GRANT statement is supported for backward compatibility only.</p> <p>Consider using the SQL-92 standard <a href="#">GRANT</a> syntax for object permissions and placing the column list before the ON clause.</p>
<p>The term integrated security allowed a SQL Server to use Windows NT Authentication mechanisms to validate logins for all connections. Standard security used SQL Server's own login validation process for all connections. Mixed security allowed login requests to be validated using either integrated or standard security.</p>	<p>The terms Windows Authentication and Mixed Mode replace integrated security and mixed security, respectively. Standard security no longer exists.</p> <p>Consider using the terms Windows Authentication and Mixed Mode rather than integrated security and mixed security. Do not refer to standard security. For more information about security modes, see <a href="#">Authentication</a>.</p>
<p>The SETUSER statement allowed a database owner to impersonate another user.</p>	<p>SETUSER is included in Microsoft® SQL Server™ 2000 for backward compatibility only, and is not recommended. This statement may no longer be supported in a future release of SQL Server.</p> <p>Consider removing all references to <a href="#">SETUSER</a>.</p>

## Installing SQL Server

### SELECT (Level 3)

SQL Server 6.x	SQL Server 2000
The FASTFIRSTROW optimizer hint caused the optimizer to use the nonclustered index if one matches the ORDER BY clause.	<p>The OPTION (FAST <i>n</i>) query hint replaces FASTFIRSTROW. However, FASTFIRSTROW is maintained for backward compatibility only.</p> <p>Consider removing all references to FASTFIRSTROW in <a href="#">SELECT</a> statements and replacing with references to OPTION (FAST <i>n</i>).</p>
The INDEX = syntax specified one or more indexes to use for a table hint.	<p>Supported for backward compatibility only.</p> <p>Consider removing all references to INDEX = and replacing (when using multiple index hints) with references to INDEX(<i>index, index...</i>) as shown in <a href="#">SELECT</a>.</p>

## Installing SQL Server

### SET SHOWPLAN (Level 3)

SQL Server 6.x	SQL Server 2000
<p>The SET SHOWPLAN statement returned output as informational messages through <b>SQLGetDiagRec</b> in ODBC, or through the message handler in DB-Library applications.</p>	<p>The SET SHOWPLAN statement is no longer supported. It has been replaced by SET SHOWPLAN_TEXT and SET SHOWPLAN_ALL. The output of SET SHOWPLAN_TEXT and SET SHOWPLAN_ALL is returned not as informational messages, but as a result set.</p> <p>Consider removing all references of SET SHOWPLAN and replacing with references to either <a href="#">SET SHOWPLAN_TEXT</a> (to display readable text) or <a href="#">SET SHOWPLAN_ALL</a> (to display output that can be parsed more easily by an application building a report of showplan output). The application needs to process the output as part of the result set, not as messages returned through the ODBC <b>SQLGetDiagRec</b> function or the DB-Library message handler.</p>

## Installing SQL Server

### System Stored Procedures (Extended) (Level 3)

SQL Server 6.x	SQL Server 2000
<p><b>xp_grantlogin</b> and <b>xp_revokelogin</b> granted or revoked SQL Server access to a Windows NT-based group or user.</p>	<p>Use <b>sp_grantlogin</b> and <b>sp_revokelogin</b> even though <b>xp_grantlogin</b> and <b>xp_revokelogin</b> are supported for backward compatibility only.</p> <p>Consider removing all references of <b>xp_grantlogin</b> and <b>xp_revokelogin</b> and replacing with references to <a href="#"><u>sp_grantlogin</u></a> and <a href="#"><u>sp_revokelogin</u></a>, respectively.</p>

## Installing SQL Server

### System Stored Procedures (System) (Level 3)

SQL Server 6.x	SQL Server 2000
Job management was called task management, and several system stored procedures allowed system administrators to create and manage tasks.	<p>The task management system stored procedures are no longer documented and are included for backward compatibility only.</p> <p>Even though Microsoft® SQL Server™ 2000 supports the task management system stored procedures (<b>sp_addtask</b>, <b>sp_droptask</b>, <b>sp_helptask</b>, <b>sp_reassigntask</b>, and <b>sp_updatetask</b>) for scheduling and managing SQL Server jobs, consider using either SQL Server Enterprise Manager or the job-related system stored procedures listed in the following table for managing jobs.</p>

The task-related stored procedures listed in the **SQL Server 6.x** column below have been replaced by the corresponding job-related stored procedures shown in the **SQL Server 2000** column.

SQL Server 6.x	SQL Server 2000
<b>sp_addtask</b>	<a href="#"><u>sp_add_job</u></a> <a href="#"><u>sp_add_jobstep</u></a> <a href="#"><u>sp_add_jobschedule</u></a> <a href="#"><u>sp_start_job</u></a>
<b>sp_droptask</b>	<a href="#"><u>sp_delete_job</u></a> <a href="#"><u>sp_delete_jobstep</u></a> <a href="#"><u>sp_delete_jobschedule</u></a>
<b>sp_helptask</b>	<a href="#"><u>sp_help_jobhistory</u></a> <a href="#"><u>sp_help_jobschedule</u></a> <a href="#"><u>sp_help_jobstep</u></a>
<b>sp_reassigntask</b>	<a href="#"><u>sp_purge_jobhistory</u></a>

	<a href="#"><u>sp_stop_job</u></a>
<b>sp_updatetask</b>	<a href="#"><u>sp_update_job</u></a> <a href="#"><u>sp_update_jobstep</u></a> <a href="#"><u>sp_update_jobschedule</u></a>

## Installing SQL Server

### Aliases (Level 4)

SQL Server 6.x	SQL Server 2000
<p>An alias allowed a user to temporarily assume the identity of another user within a database and perform actions as the aliased user. For example, the database owner could be aliased to a user so they could act as that user, if the user were on vacation.</p>	<p>Roles have replaced aliases. Because a user can belong to more than one role at a time, it is no longer necessary to assume the identity of another. Users belonging to the same roles have the same permissions automatically, assuming permissions are only applied at the role level, not the user level.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use roles instead of aliases. For more information about database roles, see <a href="#">Managing Permissions</a>.</p>

## Installing SQL Server

### Backup and Restore (Level 4)

SQL Server 6.x	SQL Server 2000
The LOAD HEADERONLY statement retrieved a result set detailing the header information from a database dump.	The result set has changed. Expect a different result set from RESTORE HEADERONLY, compared to LOAD HEADERONLY in earlier versions of SQL Server. For more information about RESTORE HEADERONLY, see <a href="#">RESTORE HEADERONLY</a> .

## Installing SQL Server

### Configuration (Level 4)

Administration scripts may have used these configuration options. For more information about configuration options, see [sp\\_configure](#) and [Setting Configuration Options](#).

SQL Server 6.x	SQL Server 2000
Administration scripts may have used the nonadvanced <b>media retention</b> option to set the number of days to retain each backup medium after it is used for a database or transaction log dump.	<b>media retention</b> is now an advanced option.  Expect different results as compared to earlier versions of Microsoft® SQL Server™. Expect the <b>media retention</b> configuration option to appear only if you have enabled the advanced configuration options of <a href="#">sp_configure</a> .

## Installing SQL Server

### **CREATE PROCEDURE (Level 4)**

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
CREATE PROCEDURE statements failed if they contained a CREATE TABLE or SELECT INTO statement creating a temporary table with the same name as a temporary table that existed at the time the CREATE PROCEDURE statement was executed.	The CREATE PROCEDURE statement succeeds.  Recode any logic that depended on the earlier behavior.

## Installing SQL Server

### Data Types (Level 4)

SQL Server 6.x	SQL Server 2000
<p>The <b>decimal</b> and <b>numeric</b> data types could use anywhere from 2 through 17 bytes to store a value, depending on the precision of the stored value.</p>	<p><b>numeric</b> and <b>decimal</b> now use 5, 9, 13, or 17 bytes of storage.</p> <p>Expect different results as compared to earlier versions of Microsoft® SQL Server™. Be sure that databases using the <b>numeric</b> or <b>decimal</b> data types have sufficient storage for the change in storage bytes.</p>
<p>Results that were too small to display, called floating point underflow, returned inconsistent results for some mathematical operators and functions.</p>	<p>SQL Server now returns 0.0 and no error message for all instances of floating point underflow.</p> <p>Because of the fixed size of floating point numbers like the <b>float</b> and <b>real</b> data types, approximate numeric data have intrinsic precision and ranges of values. In cases of floating point underflow, a result of 0.0 will be returned and no error message will be displayed. For example, the mathematical calculation of 2 to the -100.0 power would have a result 0.0.</p> <p>Expect different results as compared to earlier versions of SQL Server. Expect different results with floating point underflow with the mathematical functions or operators. For more information, see <a href="#">Using Mathematical Functions</a>.</p> <p><a href="#">+ (Add) - (Subtract)</a></p>

	<a href="#">* (Multiply)</a> <a href="#">/ (Divide)</a> <a href="#">ATN2</a> <a href="#">AVG</a> <a href="#">CONVERT</a> <a href="#">EXP</a> <a href="#">POWER</a> <a href="#">RADIANS</a> <a href="#">SUM</a>
<p>A negative second parameter equal to the number of digits in the ROUND expression returned a value of 0 for <b>integer</b>, <b>float</b>, and <b>money</b> data types. When the second parameter was negative and less than the number of digits in the ROUND expression, ROUND returned a value that rounded the right-most digit down to 0.</p>	<p>When the second parameter in the ROUND function is a negative value (for all numeric data types) that is less than the number of digits in the expression, SQL Server returns a value that is rounded up to the next digit position.</p> <p>Expect different results with the <a href="#">ROUND</a> function, compared to earlier versions of SQL Server, when the second parameter is negative.</p>
<p>The DATEADD and DATEDIFF functions returned a date value when adding or subtracting date values.</p>	<p>Direct date value addition and subtraction operations are supported for <a href="#">datetime and smalldatetime</a> using the + (Add) and -(Subtract) operators.</p> <p>For simple date arithmetic, you can also use addition ( <a href="#">+ (Add)</a> ) or subtraction ( <a href="#">- (Subtract)</a> ) instead of DATEADD and DATEDIFF.</p>

## DATEPART and SET DATEFIRST (Level 4)

SQL Server 6.x	SQL Server 2000
<p>The SET DATEFIRST setting of the DATEPART function had no effect on the <b>week</b> datepart.</p>	<p>The <b>week</b> datepart may give values different from earlier versions of Microsoft® SQL Server™. However, any difference will appear only if the SET DATEFIRST setting is not the default (the U.S. English default is 7).</p> <p>If the year provided in the DATEPART function has 366 days, a week value of 54 can be returned if the first week of the year starts on a Saturday, and the year ends on the same day of the week with the first day of the week counted from Sunday.</p> <p>When using the ISO 8601 standard, week values are always from 1 through 53, as the first week of the year is guaranteed to have a minimum of 4 days.</p> <p>Expect different results as compared to earlier versions of SQL Server. Use the default value for <a href="#">SET DATEFIRST</a> so that <a href="#">DATEPART</a> returns the expected results for the <b>week</b> datepart. Otherwise, DATEPART values will be one less than expected.</p>

## Installing SQL Server

### DBCC (Level 4)

SQL Server 6.x	SQL Server 2000
Each DBCC statement had a certain output format.	The output formats of many DBCC statements have changed.  Expect different results as compared to earlier versions of Microsoft® SQL Server™.
The DBCC PERFMON and DBCC SQLPERF statements documented SQL Server performance statistics used for studying SQL Server performance.	No longer documented. These statements may change in a future release of SQL Server.  Use the Windows 2000 System Monitor Windows NT 4.0 Performance Monitor to monitor the performance counters for SQL Server. For more information, see <a href="#">Monitoring with Windows Performance Monitor</a> .

## Installing SQL Server

### DBCS String Comparisons (Level 4)

<b>SQL Server 6.x</b>	<b>SQL Server 2000</b>
When comparing DBCS space characters, the Unicode A140 space character (U-A140) was not equal to the Unicode 0020 (U-0020) space character.	Comparisons involving the Unicode A140 space character (U-A140) are now equivalent to the Unicode 0020 (U-0020) space character.  Expect different results as compared to earlier versions of SQL Server when comparing DBCS space characters.

## Installing SQL Server

### DELETE and SELECT (Level 4)

SQL Server 6.x	SQL Server 2000
<p>Duplicate table names in the FROM clause of Microsoft® SQL Server™ version 6.0 DELETE or SELECT statement caused SQL Server to treat both table references as the same table. SQL Server discarded the reference to the second <b>authors</b> table in this SELECT example:</p> <pre>USE pubs GO SELECT * FROM authors, authors GO</pre> <p>However, if the table names specified in the FROM clause of the DELETE or SELECT were not identical, SQL Server version 6.0 treated the two table references as two different tables as in this SELECT example:</p> <pre>USE pubs GO SELECT * FROM pubs..authors, pubs.dbo.authors GO</pre>	<p>Duplicate table names in the FROM SELECT statement generate errors statements using aliases. Here is a SELECT *</p> <pre>FROM pubs..authors AS a1, p</pre> <p>USE pubs SELECT * FROM authors AS au1, author</p> <p>Expect different results as compare 6.0. Rewrite DELETE and SELECT aliases in the <a href="#">FROM</a> clause when r instance of the same table.</p>

## Installing SQL Server

### Devices (Level 4)

SQL Server 6.x	SQL Server 2000
<p>The DISK RESIZE statement altered the size of a database device.</p>	<p>The DISK RESIZE statement is supported, but may not be supported in future releases. In addition, the DISK RESIZE statement does not alter the size of the database. Instead, use ALTER DATABASE.</p> <p>Expect different results as compared to earlier versions of Microsoft® SQL Server™. Use the MODIFY FILE clause of the <a href="#">ALTER DATABASE</a> statement to alter the size of a database.</p>

## Installing SQL Server

### Functions (Level 4)

SQL Server 6.x	SQL Server 2000
<p>The @@DBTS global variable was incremented any time any page in the database was modified in any way.</p>	<p>The value returned by the @@DBTS function changes only if a row containing a <b>timestamp</b> column is modified.</p> <p>Expect different results as compared to earlier versions of Microsoft® SQL Server™ when using <a href="#">@@DBTS</a>.</p>

## Installing SQL Server

### Global Variables (Level 4)

Pre-SQL Server 7.0	SQL Server 7.0
Global variables were system-supplied, predeclared variables that were distinguished from local variables by having two at symbols (@@) preceding their names.	Transact-SQL global variables are a form of function and are now referred to as functions. For more information, see <a href="#">Functions</a> .

## Installing SQL Server

### ODBC (Level 4)

SQL Server 6.x	SQL Server 2000
<p>In the version 2.65 ODBC driver, the long-running query interval, specified by calling <b>SQLSetConnectOption</b> with the driver-specific connection option <code>SQL_COPT_SS_PERF_QUERY_INTERVAL</code>, was specified in seconds.</p>	<p>The <code>SQL_COPT_SS_PERF_QUERY</code> value is specified in milliseconds. Expect different results as compared to earlier versions of Microsoft® SQL Server. Multiply the value of <code>SQL_COPT_SS_PERF_QUERY</code> by 1,000 to convert the number of milliseconds. For more information on <code>SQL_COPT_SS_PERF_QUERY</code>, see <a href="#">SQLSetConnectAttr</a>.</p>
<p>For earlier versions of the ODBC SQL Server driver, messages from consecutive PRINT, RAISERROR, DBCC, or similar statements (in a batch or stored procedure) were combined into a single result set.</p>	<p>For the ODBC SQL Server 3.51-4 driver (included with SQL Server 2000), messages from consecutive PRINT, RAISERROR, DBCC, or similar statements (in a batch or stored procedure) are returned as a separate result set for each statement. Expect different results as compared to earlier versions of SQL Server. Call <a href="#">SQLMoreResults</a> to process the results from each statement.</p>
<p>Earlier versions of the ODBC SQL Server driver returned <code>SQL_SUCCESS</code> when executing a searched UPDATE or DELETE statement that affects no rows (using <b>SQLExecute</b>, <b>SQLExecDirect</b>, or <b>SQLParamData</b>). <b>SQLRowCount</b> returned zero.</p>	<p>When an ODBC version 3.x application uses the ODBC SQL Server 3.5 driver with this release, it returns <code>SQL_SUCCESS</code> when executing a searched UPDATE or DELETE statement that affects no rows (using <b>SQLExecute</b>, <b>SQLExecDirect</b>, or <b>SQLParamData</b>). <b>SQLRowCount</b> returns zero. Expect different results as compared to earlier versions of SQL Server.</p>

Expect different results as compared to earlier versions of SQL Server. Handle SQL\_NO\_DATA appropriately.

## Installing SQL Server

### Rebuilding the master Database (Level 4)

SQL Server 6.x	SQL Server 2000
Executing SQL Server Setup rebuilt the <b>master</b> database.	No longer supported. SQL Server includes the Rebuild Master ( <b>rebuildm</b> ) utility.  Use the <a href="#">Rebuild Master (rebuildm) Utility</a> located in the x:\Program Files\Microsoft SQL Server\80\Tools\Binn folder to rebuild the <b>master</b> database.

## Installing SQL Server

### **Rebuilding the Registry (Level 4)**

**Note** In Microsoft® SQL Server™ 2000, this utility is replaced by the setup option, Registry Rebuild.

<b>Pre-SQL Server 7.0</b>	<b>SQL Server 7.0</b>
Executing this SQL Server Setup statement rebuilt the registry: setup/t RegistryRebuild = On.	No longer supported. Instead, use Setup to rebuild the registry.

## Installing SQL Server

### Replication (Level 4)

SQL Server 6.x	SQL Server 2000
Subscriptions to one or more articles in a publication were created either through SQL Server Enterprise Manager or through the appropriate system stored procedures.	SQL Server Enterprise Manager no longer allows subscription to one or more articles. Subscribing to one or more articles of a publication can be done only by using the appropriate replication system stored procedures.  Expect different results as compared to earlier versions of SQL Server. Use the replication system stored procedures to subscribe to one or more articles of a publication. When using SQL Server Enterprise Manager, it is necessary to subscribe to the entire publication.

Replication functions differently between SQL Server 2000 and SQL Server version 6.5 servers. In addition, SQL Server 2000 offers enhanced scripting ability after your replication topology is created in the user interface. This enhanced scripting allows mass implementation of replication topology with a minimum of time and effort.

## Installing SQL Server

### Security (Level 4)

SQL Server 6.x	SQL Server 2000
The SYSTEM_USER function returned nulls for any Microsoft Windows NT® login.	The appropriate domain and login names are returned if Windows Authentication is used with the SYSTEM_USER function.  Expect a different result, as compared to earlier versions, when using <a href="#">SYSTEM_USER</a> with Windows Authentication.

## Installing SQL Server

### SELECT (Level 4)

SQL Server 6.x	SQL Server 2000
<p>The left outer join (*=) and right outer join (=*) operators were used in SELECT statements to produce left and right outer joins, respectively.</p>	<p>The SQL-92-standard syntax of LEFT OUTER JOIN and RIGHT OUTER JOIN is preferred. However, join operators supported in earlier versions of Microsoft® SQL Server™ are supported.</p> <p>It is recommended that you remove all references of the left outer join (*=) and right outer join (=*) operators in all <a href="#">SELECT</a> statement <a href="#">FROM</a> clauses and replace with references to the SQL-92-standard syntax RIGHT OUTER JOIN and LEFT OUTER JOIN. Future versions of SQL Server will support only the SQL-92-standard syntax.</p>

## Installing SQL Server

### Triggers and System Stored Procedures (System) (Level 4)

SQL Server 6.x	SQL Server 2000
Returned values were not always correct for <b>text</b> or <b>image</b> columns in either the <b>inserted</b> or <b>deleted</b> tables when either table was used in a CREATE TRIGGER statement.	<p>NULL values are returned for <b>text</b> or <b>image</b> column references in the <b>inserted</b> or <b>deleted</b> tables in CREATE TRIGGER.</p> <p>Expect different results as compared to earlier versions of SQL Server. Remove all references to either the <b>text</b> or <b>image</b> columns of the <b>inserted</b> or <b>deleted</b> tables when used in <a href="#">CREATE TRIGGER</a> statements.</p>
Direct recursion of triggers (the ability of a trigger to call itself) was not supported, but indirect recursion was allowed.	<p>Direct trigger recursion is enabled with the RECURSIVE_TRIGGERS option of ALTER DATABASE. Indirect recursion is enabled with the <b>nested triggers</b> configuration option.</p> <p>Expect different results as compared to earlier versions of SQL Server.</p>
Server-to-server communication existed between version 4.x and version 6.x servers when initialized by either side.	<p>Version 4.x or 6.x servers can communicate with SQL Server 2000 servers. However, server-to-server communication is not supported from SQL Server 2000 servers to 4.x servers.</p> <p>Expect different results as compared to earlier versions of SQL Server. Upgrade the 4.x server to either SQL Server 6.x or SQL Server 2000</p>
<b>sp_tableoption</b> set option values for user-defined tables, including	The Insert Row Locking (IRL) parameters in <b>sp_tableoption</b> are not

<p>the use of Insert Row Locking (IRL).</p>	<p>supported but have been replaced with complete row-level locking.</p> <p>Expect different results as compared to earlier versions of SQL Server. Remove all references to IRL actions implemented using <a href="#">sp_tableoption</a> and use the built-in row-level locking of SQL Server 2000 instead. Applications calling <b>sp_tableoption</b> should continue to work properly; the IRL parameters will be ignored.</p>
<p>The <b>@message</b> parameter of <b>xp_readmail</b> was <b>varchar(255)</b>. The <b>@message</b> and <b>@query</b> parameters of <b>xp_sendmail</b> were <b>varchar(255)</b>.</p>	<p>The <b>@message</b> and <b>@query</b> parameters are now <b>varchar(8000)</b>.</p> <p>Expect differences in behavior as compared to earlier versions of SQL Server when using the <b>@message</b> parameter of <a href="#">xp_readmail</a> and the <b>@message</b> and <b>@query</b> parameters of <a href="#">xp_sendmail</a>.</p>

## Installing SQL Server

### UPDATE (Level 4)

SQL Server 6.x	SQL Server 2000
<p>In Microsoft® SQL Server™ version 6.0, the following UPDATE statement, using two different table aliases for the same base table, was allowed:</p> <pre>CREATE TABLE t1 (c1 int) GO INSERT t1 VALUES (1) INSERT t1 VALUES (2) GO UPDATE t1 SET c1 = 50 FROM t1 a1, t1 a2 WHERE a1.c1 = 1 AND a2.c1 = 2 GO</pre>	<p>Syntax no longer supported UPDATE keyword. The UPDATE statement is now:</p> <pre>UPDATE a1 SET c1 = 50 FROM t1 a1, t1 a2 WHERE a1.c1 = 1 AND a2.c1 = 2</pre> <p>Expect differences in behavior.</p>
<p>This UPDATE statement with table and alias references worked.</p> <pre>USE pubs GO UPDATE titles SET t.ytd_sales = t.ytd_sales + s.qty FROM titles t, sales s WHERE t.title_id = s.title_id AND s.ord_date = (SELECT MAX(sales.ord_date) FROM sales) GO</pre>	<p>The alias specified after the SET keyword must be changed to 65 for t.</p> <p>Here is the same UPDATE statement:</p> <pre>USE pubs GO UPDATE t SET t.ytd_sales = t.ytd_sales + s.qty FROM titles t, sales s WHERE t.title_id = s.title_id AND s.ord_date = (SELECT MAX(sales.ord_date) FROM sales) GO</pre>

Expect differences in behavior  
different table references for  
statement and the SET key

## Installing SQL Server

### Utilities (Level 4)

SQL Server 6.x	SQL Server 2000
<p><b>isql/w</b> used DB-Library. The SQL-92 settings, like ANSI_WARNINGS, were set off, by default.</p>	<p>SQL Query Analyzer uses the SQL Server ODBC driver, which, by default, sets these SQL-92 options on: SET ANSI_WARNINGS, SET ANSI_PADDING, and SET ANSI_NULLS. Any errors returned are formatted as ODBC errors rather than DB-Library errors.</p> <p>Expect different results as compared to earlier versions of Microsoft® SQL Server™. Expect different results with <a href="#">SQL Query Analyzer</a>, compared to the <a href="#">isql utility</a> or the SQL Server 6.5 version of <b>isql/w</b>.</p>
<p><b>isql/w</b> used the ANSI-ISO code pages. When connected to a server using OEM code page 850 or 437, <b>ANSI to OEM character translation</b> had to be explicitly enabled. Otherwise, data with extended characters appeared garbled.</p>	<p>The SQL Server 2000 ODBC driver automatically detects the need for and sets up automatic ANSI to OEM conversion. In addition, when SQL Query Analyzer connects, automatic detection is enabled.</p> <p>Expect a change in behavior when using SQL Query Analyzer with international or extended characters.</p>

For additional information about changes to SQL Query Analyzer, see the discussion "Default Connection Option Settings in SQL Query Analyzer" in [SQL Server 2000 and SQL Server version 7.0](#).