Samples
Samples

SQL Server 2000 provides programming samples that demonstrate how to use a number of different technologies with a variety of programming languages. You can install these as part of a custom installation.

You can use the following table to view all available samples, or you can select a specific category by clicking the Technology down arrow and making a specific selection (for example, Replication). You can also sort the table by clicking a column heading (for example, Language). To view detailed information about a specific sample, click the text in the Sample column.

ADO Employee MS-ITS:adosql.chm::/adoprg04_9nc9.htm Visual Basic Displays long data types using the Employees table in the Northwind database.
ADO Employee MS-ITS:/adosql.chm::/adoprg04_1sa1.htm C++ Displays long data types using the Employees table in the Northwind database.
ADO FORXML MS-ITS:adosql.chm::/adoprg04_92gc.htm XML Builds an Active Server Page (ASP) that retrieves result sets into streams.
ADO Intro MS-ITS:adosql.chm::/adoprg04_5iw7.htm Visual Basic Connects to and queries a SQL Server database. Also shows error handling.
ADO OpenXML MS-ITS:adosql.chm::/adoprg04_15ik.htm XML Builds an Active Server Page (ASP) that maps an XML schema to a relational schema.
ADO Xpath MS-ITS:adosql.chm::/adoprg04_21vd.htm XML Builds an Active Server Page (ASP) that performs an xpath query.
ADO Web MS-ITS:adosql.chm::/adoprg04_85ym.htm Web Builds Active Server Page (ASP) that interacts with instance of SQL Server.
Analysis Services AspAdoComplex MS-ITS:olapdmpr.chm::/prsamples_1rji.htm VBScript Creates and manipulates a query-based cube.
Analysis Services AspAdoCubeDoc MS-ITS:olapdmpr.chm::/prsamples_6p7y.htm VBScript Retrieves and manipulates a cube schema.
Analysis Services AspAdoCubeTree MS-ITS:olapdmpr.chm::/prsamples_6p7y.htm VBScript, JScript Retrieves and manipulates a cube schema.
Analysis Services AspAdoSimple MS-ITS:olapdmpr.chm::/prsamples_1rji.htm VBScript Creates and manipulates a query-based cube.
Analysis Services CppOlapDemo MS-ITS:olapdmpr.chm::/prsamples_1rji.htm C++ Creates and manipulates a query-based cube.
Analysis Services VbAdoComplex MS-ITS:olapdmpr.chm::/prsamples_1rji.htm Visual Basic Creates and manipulates a
Library (ATL) ATL template for a DTS custom transformation. DTS DTSActiveScriptTask MS-ITS:dtsprog.chm::/dtspsamp_76p7.htm Visual Basic Runs a VBScript as part of a DTS Task. DTS DTSApplObject MS-ITS:dtsprog.chm::/dtspsamp_91ys.htm Visual Basic Uses DTS application object on the local server. DTS DTSApplObject MS-ITS:dtsprog.chm::/dtspsamp_4a70.htm Visual Basic Gets information from the DTS Application object. DTS DTSBulkInsertTask MS-ITS:dtsprog.chm::/dtspsamp_14iz.htm Visual Basic Uses DTS to do a Bulk Insert from a flat text file. DTS DTSCopy MS-ITS:dtsprog.chm::/dtspsamp_0k55.htm C++ Copies a single source column of any simple type into a destination column of the same type. DTS DTSCopyDatabase MS-ITS:dtsprog.chm::/dtspsamp_5npn.htm Visual Basic Uses the DTS TransferObjectsTask object to copy a database. DTS DTSExecProcess MS-ITS:dtsprog.chm::/dtspsamp_9ik3.htm Visual Basic Runs a Win32 application from a DTS task. DTS DTSExecSQLTask MS-ITS:dtsprog.chm::/dtspsamp_2ip7.htm Visual Basic Executes a SQL statement while running a DTS package. DTS DTSExecutePackage MS-ITS:dtsprog.chm::/dtspsamp_6jvp.htm Visual Basic Executes programmatically a DTS package that has been saved to structured storage (i.e. .dts file). DTS DTSEXmp1 MS-ITS:dtsprog.chm::/dtspsamp_5wvn.htm Visual Basic Copies data from a source table to a destination table. DTS DTSEXmp2 MS-ITS:dtsprog.chm::/dtspsamp_7zcc.htm Visual Basic Copies database to Microsoft Excel spreadsheet and creates pivot table; also creates a DTS package. DTS DTSEXmp3 MS-ITS:dtsprog.chm::/dtspsamp_3ur7.htm Visual Basic Creates a package supporting multiple sources and destinations. DTS DTSTFTPTask MS-ITS:dtsprog.chm::/dtspsamp_6ai3.htm Visual Basic Uses DTS to copy files external to SQL Server from a source to a destination. DTS DTSPackageInfo MS-ITS:dtsprog.chm::/dtspsamp_5rov.htm Visual Basic Gets information using the DTS Application GetPackageInfos method. DTS DTSTStrings MS-ITS:dtsprog.chm::/dtspsamp_47n7.htm C++ Reformats two source columns. DTS DTSTask MS-ITS:dtsprog.chm::/dtspsamp_72qz.htm C++ Creates and registers a CreateProcessTask COM object. DTS DTSTransferObjectsTask MS-ITS:dtsprog.chm::/dtspsamp_65yj.htm Visual Basic Uses DTS to transfer various types of SQLServer objects from a pubs database to a pubs2 database. DTS DTSTskGVUpdate MS-ITS:dtsprog.chm::/dtspsamp_12lh.htm C++ Displays and allows the user to update the value of a global variable. DTS DTSTskPropIcon MS-
C++ Displays a message whose text is specified as a custom task property. DTS DTSvbpkg MS-ITS Displays and registers a custom Transformation object. DTS FoodMart2000 MS-ITS DTSxform MS-ITS:dtsprog.chm::/dtspsamp_9f3i.htm C++ Creates and registers a custom Transformation object. DTS FoodMart2000 MS-ITS DTSxform MS-ITS:dtsprog.chm::/dtspsamp_5ns3.htm Visual Basic Builds and executes a DTS package. DTS DTSxform MS-ITS:dtsprog.chm::/dtspsamp_74h1.htm C++ Creates and registers a custom Transformation object. DTS FoodMart2000 MS-ITS DTSxform MS-ITS:dtsprog.chm::/dtspsamp_0f8z.htm C++ Creates and registers a custom Transformation object. DTS Pub2Pubs MS-ITS:dtsprog.chm::/dtspsamp_2m7n.htm Visual Basic Copies the authors table from the 'pubs' database to the 'pubs2' database, while doing some operations on various fields. Embedded SQL Edlib MS-ITS:esqlforc.chm::/ec_6_eex_00_9o36.htm C Uses GET CONNECTION to obtain the DB-Library DBPROCESS connection pointer. Embedded SQL Embedded SQL for C Examples MS-ITS:esqlforc.chm::/ec_6_eex_00_6376.htm C 8 examples using C to connect to a specified instance of Microsoft SQL Server and execute a series of Embedded SQL statements. Embedded SQL Genchar MS-ITS:esqlforc.chm::/ec_6_eex_00_4uya.htm C Issues a SELECT statement query from the authors table in the pubs database. Embedded SQL Genwin MS-ITS:esqlforc.chm::/ec_6_eex_00_2a7m.htm C Issues a SELECT statement query from the authors table in the pubs database. Extended Stored Procedure xp-dblib MS-ITS:odssql.chm::/ods_6_sam_01_4xia.htm C Uses DB Library to open bound connection to same SQL Server that called the extended stored procedure. Extended Stored Procedure xp-hello MS-ITS:odssql.chm::/ods_6_sam_01_060f.htm C Displays "hellow world" in output row. Extended Stored Procedure xp-ODBC MS-ITS:odssql.chm::/ods_6_sam_01_9jkz.htm C Uses ODBC to open bound connection to same SQL Server that called the extended stored procedure. Extended Stored Procedure xp-param MS-ITS:odssql.chm::/ods_6_sam_01_4yel.htm C Accepts parameters from Transact-SQL script and reports values. MS DTC DB Library MS-ITS:samples.chm::/samp_msdtc_5a79.htm C++ Shows how to use MS-DTC in a C++ program. MS DTC ODBC MS-ITS:samples.chm::/samp_msdtc_5tnp.htm C++ Shows how to use MS-DTC in a C++ program, using ODBC. MS DTC T-SQL MS-ITS:samples.chm::/samp_msdtc_26hx.htm Transact-SQL Shows how to use MS-DTC in a Transact-SQL script. ODBC bcp MS-
ITS:odbcsql.chm::/od_odbcamp_4pm8.htm C++ Uses bulk copy functions with ODBC driver. ODBC Compute MS-ITS:odbcsql.chm::/od_odbcamp_7yb7.htm C++ Executes SELECT statement, then determines shape of each result set generated. ODBC Loaddata MS-ITS:odbcsql.chm::/od_odbcamp_74rl.htm C++ Loads data into SQL Server table. ODBC Mfcperf MS-ITS:odbcsql.chm::/od_odbcamp_7xly.htm C++ Tunes MFC ODBC database classes for performance, and captures and interprets Microsoft SQL Server ODBC driver performance data. ODBC Trans MS-ITS:odbcsql.chm::/od_odbcamp_8037.htm C++ Uses cursors to enable concurrent data access. OLE Automation Getnpv MS-ITS:samples.chm::/samp_oleauto_04md.htm Visual Basic Retrieves net present value. OLE Automation Loopback (DAO) MS-ITS:samples.chm::/samp_oleauto_7x45.htm Visual Basic Retrieves contents of the input table name provided to the DAO table and returns a tabular array equivalent to the result set returned from SQL Server. OLE Automation Loopback (RDO) MS-ITS:samples.chm::/samp_oleauto_5lyd.htm Visual Basic Uses RDO to call SQL Server and retrieve the contents of the authors table; returns a tabular array equivalent to the result set returned from SQL Server. OLE Automation Traverse MS-ITS:samples.chm::/samp_oleauto_5t2d.htm Transact-SQL Traverses an object hierarchy. Replication Deflt_sp MS-ITS:replprog.chm::/rp_replsamp_446q.htm Transact-SQL Builds a custom stored procedure resolver. Replication Distsamp MS-ITS:replprog.chm::/rp_replsamp_2pf2.htm C++ Shows how to include the SQL Merge and SQL Distribution controls in a custom application. Replication Repldts MS-ITS:replprog.chm::/rp_replsamp_3l0j.htm Visual Basic Creates a transformation subscription. Replication Replsamp MS-ITS:replprog.chm::/rp_replsamp_5uwe.htm HTML Shows how to include the SQL Merge and SQL Distribution controls in a custom application. Replication Replsamp.vbp MS-ITS:replprog.chm::/rp_replsamp_7yr2.htm Visual Basic Shows how to include the SQL Merge and SQL Distribution controls in a custom application. Replication Samppub MS-ITS:replprog.chm::/rp_replsamp_2mr3.htm Transact-SQL Configures a third-party publication and distributor. Replication Subspres MS-ITS:replprog.chm::/rp_replsamp_6p5x.htm C++ Builds a custom stored procedure resolver that executes the stored procedure at the Subscriber. Replication xp_enumntusers MS-ITS:replprog.chm::/rp_replsamp_3rsj.htm C++ Generates multiple merge dynamic snapshot jobs. SQL-DMO AxSQLDMOCtl
MS-ITS:sqldmo.chm::/dmoref_ex01_3uy4.htm Visual Basic Creates a UserControl. SQL-DMO BackRestEvents MS-ITS:sqldmo.chm::/dmoref_ex01_0kix.htm Visual Basic Backs up and restores a database. SQL-DMO BackRestEvents MS-ITS:sqldmo.chm::/dmoref_ex01_01gp.htm C++ Backs up and restores a database. SQL-DMO BackupDevice MS-ITS:sqldmo.chm::/dmoref_ex01_325h.htm Visual Basic Uses BackupDevice object to add and remove a backup device. SQL-DMO CreateDatabase MS-ITS:sqldmo.chm::/dmoref_ex01_0bs5.htm Visual Basic Creates a database. SQL-DMO CreateTable MS-ITS:sqldmo.chm::/dmoref_ex01_5w11.htm Visual Basic Uses SQL-DMO objects to create and alter tables. SQL-DMO DMOExplorer MS-ITS:sqldmo.chm::/dmoref_ex01_53ci.htm Visual Basic Walks the DMO object model and shows the values in it. SQL-DMO Dmoping MS-ITS:sqldmo.chm::/dmoref_ex01_82pl.htm C++ Determines version of SQL Server instance. SQL-DMO Enums MS-ITS:sqldmo.chm::/dmoref_ex01_8roz.htm Visual Basic Enumerates SQL Server enumeration methods and recordsets. SQL-DMO Explore MS-ITS:sqldmo.chm::/dmoref_ex01_5a3p.htm Visual Basic Displays contents of collections and properties of objects in SQL-DMO object tree. SQL-DMO Idxtest MS-ITS:sqldmo.chm::/dmoref_ex01_3cc4.htm Visual Basic Tests optimization strategies for stored procedures and views. SQL-DMO Login MS-ITS:sqldmo.chm::/dmoref_ex01_0b72.htm Visual Basic Locates available SQL Servers and logs in to them. SQL-DMO Registry MS-ITS:sqldmo.chm::/dmoref_ex01_7gxl.htm Visual Basic Finds registry information about a specified SQL Server. SQL-DMO Service MS-ITS:sqldmo.chm::/dmoref_ex01_04yt.htm Visual Basic Checks status of, starts, and stops the service. SQL-DMO Smartptr MS-ITS:sqldmo.chm::/dmoref_ex01_07oy.htm C++ Uses Microsoft Visual C++ features to optimize size of program and speed of development. SQL-DMO Soc MS-ITS:sqldmo.chm::/dmoref_ex01_40tf.htm C Shows how to use C for SQL-DMO development. SQL-DMO Soccpp MS-ITS:sqldmo.chm::/dmoref_ex01_74kw.htm C++ Shows how to use C++ for SQL-DMO development. SQL-DMO SQLScripts MS-ITS:sqldmo.chm::/dmoref_ex01_13sj.htm Visual Basic Shows how to generate SQL scripts to regenerate SQL Server objects. SQL-DMO VerifyBackup MS-ITS:sqldmo.chm::/dmoref_ex01_3t80.htm Visual Basic Finds backup devices and verifies backup set. SQL-NS Dbprop MS-
Displays Properties dialog for Pubs database. SQL-NS Dump tree Dumps all objects in the SQL namespace tree. SQL-NS Browse Visual Basic Displays browser for viewing and manipulating SQL namespace. SQL-NS Dbprop Visual Basic Displays Properties dialog for Pubs database. User Defined Functions CustNamesInRegion Transact-SQL Creates an in-line user-defined function in the Northwind database to view customer information for specific regions from the Customers table. User Defined Functions FindReports Transact-SQL Queries the Employees table in the Northwind database and shows all direct and indirect reports. User Defined Functions LargeFreight Transact-SQL Creates a user-defined function that queries the Shippers and Orders tables in the Northwind database and returns orders with freight greater than a certain value. User Defined Functions MyDate Transact-SQL Changes the format in which the date is returned. Utilities Bii Executable utility Mimics the bcp utility, but adds additional functionality. Utilities Pbalance C++ Displays process information for instances of Microsoft(r) SQL Server(tm) 2000 running on a local computer, and manages process priorities. Virtual Backup Device Mprocess C++ Demonstrates a multiple stream Backup or Restore, in which each stream is handled by a secondary process. Virtual Backup Device Mthread C++ Demonstrates a multiple stream Backup or Restore from a single process. Virtual Backup Device Osimple C++ Demonstrates how to backup or restore a database across an ODBC connection. Virtual Backup Device Simple C++ Demonstrates how to backup or restore a database. Virtual Backup Device Snapshot C++ Demonstrates how to backup or restore a database using snapshot extensions. XML XMLDemo XML Demonstrates Microsoft(r) SQL Server(tm) 2000 support for XML. XML XMLStartup XML Demonstrates features that make SQL Server an XML-enabled database server.
New and updated SQL Server programming samples are available from the Microsoft Web site. On the Resources menu, click Code Center.

Note When using the sample programs, ensure that hard-coded values, such as user name and password, are modified as necessary.
Samples
User-Defined Function Samples

These four Transact-SQL samples show how to use user-defined functions. Earlier versions of Microsoft® SQL Server™ supplied only built-in functions defined as part of the Transact-SQL language. Built-in functions operate as defined in the Transact-SQL Reference and cannot be modified. Microsoft® SQL Server™ 2000 introduces user-defined functions that allow you to define and modify your own Transact-SQL functions.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Udf_CustNamesInRegion</td>
<td>Transact-SQL sample. This script creates an in-line user-defined function in the Northwind database to view customer information for specific regions from the Customers table.</td>
</tr>
<tr>
<td>Udf_FindReports</td>
<td>Transact-SQL sample. This script creates a function that queries the Employees table in the Northwind database and shows all direct and indirect reports.</td>
</tr>
<tr>
<td>Udf_LargeFreight</td>
<td>Transact-SQL sample. This script creates a user-defined function that queries the Shippers and Orders tables in the Northwind database and returns orders with freight greater than a certain value.</td>
</tr>
<tr>
<td>Udf_MyDate</td>
<td>Transact-SQL sample. This script creates a single function that changes the format in which the date is returned. The script can be used on any table in the database.</td>
</tr>
</tbody>
</table>

To install the samples during SQL Server installation

1. On the Setup Type page, select Custom.

2. On the Select Components page, under Components, select Code Samples.
Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_m misc.exe, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc.

**Default Location**

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc\Udf

**Running the Samples**

Instructions are included in Userdefinedfunc.doc, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc\Udf.

**See Also**

[Samples](#)
Samples
Utility Samples

The following samples illustrate Microsoft® SQL Server™ 2000 utility development.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bii</td>
<td>Utility sample. Stand-alone console application that mimics the bcp utility that ships with Microsoft® SQL Server™ 2000.</td>
</tr>
<tr>
<td>Pbalance</td>
<td>C++ sample. Displays instance names, process identifiers, CPU-time as a percentage, high and low thresholds of the CPU-time, and the process priorities of instances of Microsoft® SQL Server™ 2000 running on the local computer.</td>
</tr>
</tbody>
</table>

**To install the samples during SQL Server installation**

1. On the Setup Type page, select **Custom**.

2. On the Select Components page, under **Components**, select **Code Samples**.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_utils.exe, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Utils.

**Prerequisites**

C++ samples require Microsoft Visual C++ version 6.0.
Samples
**Bii**

This sample is a stand-alone console application that mimics the **bcp** utility that ships with Microsoft® SQL Server™ 2000. Unlike the **bcp** utility, the **bii** utility detects when an image field is the destination database field and the input is a filename that can be located in the file system. In this case, instead of loading the filename string into the image field, the program opens the file, loads the data, and inserts the file contents in the image data type field.

**Default Location**

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Utils\bii

**Running the Sample**

To run the sample:

1. Open SQL Query Analyzer, connect to an instance of SQL Server, and run the Createtb.sql query.

2. Open a Command Prompt window and change the current directory to C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Utils\bii.

3. Enter the following command (The `-v` option is optional; it displays current status.):
   
bii testinsert.bii -Syourservername -Dyourdbname -Usqlusername -Psqlpassword -t, -v

**See Also**

[Utility Samples](#)
Samples
**Pbalance**

This sample displays instance names, process identifiers, CPU-time as a percentage, high and low thresholds of the CPU-time, and the process priorities of instances of Microsoft® SQL Server™ 2000 running on the local computer. The sample also manages process priorities and gives users the option of setting them manually.

You must install Microsoft Platform Software Development Kit (SDK) to compile and run this sample.

**Default Location**

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Utils\pbalance

**Running the Sample**

1. Open pbalance.dsw in Microsoft® Visual C++® 6.0 and compile the program.

2. Run pbalance.exe on your local machine. The application will display the information of the running instances.

3. To set the timer interval for collecting data, select any running instance in the list box and click **Set** (or double click any running instance name). This displays the Change Setting dialog box. On the upper-right corner, enter the new time interval value in the edit box and then click **OK**.

4. To set high and low CPU-time thresholds of a running instance, select the running instance in the list box and click **Set** (or double click any running instance name). This displays the Change Setting dialog box. Enter the new high and low CPU-time thresholds values in the edit boxes. You can also set the process priority of the running instance.
here. To do this, select the new process priority in the combo box. Click **OK** to make the new settings effective.

**Remarks**

The sample extracts the CPU real time (the percentage) consumed by each running instance once in every fixed time interval. The default value of this time interval is 5 seconds. Users have the option to set this time interval to a different value. If the CPU-time consumed by a running instance is higher than the high threshold or lower than the low threshold, the application then lowers or raises the process priority by one level. If the priority is already at the top or bottom level and cannot be lowered or raised, the priority will then remain the same. Users also have the option to set the high and low thresholds.

**See Also**

[Utility Samples](#)
Samples
Virtual Backup Device Samples

The following samples illustrate Microsoft® SQL Server™ 2000 Virtual Backup Device application development.

| Sample   | Description                                                                 |
|----------|-----------------------------------------------------------------------------|---|
| Mprocess | Visual C++ sample. Demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of SQL Server 2000. |
| Mthread  | Visual C++ sample. Demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of SQL Server 2000. |
| Osimple  | Visual C++ sample. Demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of SQL Server 2000 with ODBC. |
| Simple   | Visual C++ sample. Demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of SQL Server 2000. |
| Snapshot | Visual C++ sample. Demonstrates how to backup or restore the pubs sample database with snapshot by using the Virtual Device Interface feature of SQL Server 2000, and handles snapshot extensions. |

To install the samples during SQL Server installation

1. On the **Setup Type** page, select **Custom**.

2. On the **Select Components** page, under **Components**, select **Development Tools**.
3. Under **Sub-Components**, select **Backup/Restore API**.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_backup.exe, located at
C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup.

**Note**  The SQL Server 2000 Virtual Backup Device Specification, which contains detailed information about Virtual Backup Device application development, is also installed as Vbackup.chm.

**Prerequisites**

C++ samples require Microsoft Visual C++ version 6.0.

**See Also**

[Samples](#)
Samples
Mprocess

This Microsoft® Visual C++® console sample demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of Microsoft® SQL Server™ 2000. It demonstrates a multiple stream Backup or Restore, in which each stream is handled by a secondary process.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup\Mprocess

Running the Sample

Open the mprocess.dsw in Microsoft Visual C++ 6.0 and compile this program. Make sure to do the following before compiling.

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included:
   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where mprocess.exe is located. Run the program by entering "mprocess [b|r] [nStreams]".

The program requires two command line parameters. The first parameter can be one of the following:
   - b: perform a backup
- r: perform a restore

The second parameter gives the number of streams to use. It can be an integer from 1 to 32.

**Remarks**

This sample is designed for use with Mixed Mode security. Comments explaining how to implement Windows Authentication mode are provided in the code.

**See Also**

[Virtual Backup Device Samples]
Samples
Mthread

This Microsoft® Visual C++® console sample demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of Microsoft® SQL Server™ 2000. It demonstrates a multiple stream Backup or Restore from a single process.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup\Mthread

Running the Sample

Open the mthread.dsw in Microsoft Visual C++ 6.0 and compile this program.

Make sure to do the following before compiling:

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included:

   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where mthread.exe is located. Run the program by entering "mthread [b|r] [nStreams]".

The program requires two command line parameters. The first parameter can be one of the following:

- b: perform a backup
The second parameter, \textit{nStreams}, specifies the number of streams to use. It can be an integer from 1 to 32.

**Remarks**

This sample is designed for use with Mixed Mode security. Comments explaining how to implement Windows Authentication mode are provided in the code.

**See Also**

[Virtual Backup Device Samples](#)
Samples
Osimple

This Microsoft® Visual C++® console sample demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of Microsoft® SQL Server™ 2000. This sample extends the Simple sample to use an ODBC connection.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup\Osimple

Running the Sample

Open osimple.dsw in Microsoft® Visual C++® 6.0 and compile this program. Make sure to do the following before compiling:

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included:
   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where osimple.exe is located. Run the program by entering "osimple [b|r]".

The program requires one command line parameter. This parameter can be one of the following:

- b: perform a backup
• r: perform a restore

See Also

Virtual Backup Device Samples
Samples
Simple

This Microsoft® Visual C++® console sample demonstrates how to backup or restore the pubs sample database by using the Virtual Device Interface feature of Microsoft® SQL Server™.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup\Simple

Running the Sample

Open the simple.dsw in Microsoft® Visual C++® 6.0 and compile this program. Make sure to do the following before compiling.

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included:
   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where simple.exe is located. Run the program by entering "simple [b|r]".

The program requires one command line parameter. This parameter can be one of the following:

- b: perform a backup
• r: perform a restore

Remarks

This sample is designed for use with Mixed Mode security. Comments explaining how to implement Windows Authentication mode are provided in the code.

See Also

Virtual Backup Device Samples
Samples
Snapshot

This Microsoft® Visual C++® console sample demonstrates how to backup or restore the pubs sample database with snapshot by using the Virtual Device Interface feature of Microsoft SQL Server™ 2000. This sample extends the Osimple sample to handle the snapshot extensions. The ability to take or mount snapshots must be implemented before this sample is truly functional.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Backup\Snapshot

Running the Sample

Open snapshot.dsw in Microsoft® Visual C++® 6.0 and compile this program. Make sure to do the following before compiling:

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included:

   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include

   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where snapshot.exe is located. Run the program by entering "snapshot [b|r]".

The program requires one command line parameter. This parameter can be one of the following:
- b: perform a backup

- r: perform a restore

**See Also**

[Virtual Backup Device Samples](#)
Samples
XML Samples

The following samples illustrate Microsoft® SQL Server™ 2000 XML development.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLDemo</td>
<td>XML Sample. Demonstrates Microsoft® SQL Server™ 2000 support for XML functionality.</td>
</tr>
<tr>
<td>XMLStartup</td>
<td>XML Sample. Demonstrates a combination of these features makes SQL Server 2000 an XML-enabled database server</td>
</tr>
</tbody>
</table>

To install the samples during SQL Server installation

1. On the Setup Type page, select Custom.

2. On the Select Components page, under Components, select Code Samples.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_xml.exe, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Xml.

See Also

Samples
Samples
**XMLDemo**

This sample demonstrates Microsoft® SQL Server™ 2000 support for XML functionality.

**Default Location**

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Xml\XMLDemo

**Running the Sample**

To install virtual roots:

1. On the *Start* menu, point to *Programs/Microsoft SQL Server*, and then click *Configure SQL XML Support in IIS*.

2. Create a new virtual directory named "Demos" to access the *Northwind* database.

3. Configure these three virtual names using a local path setting:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dbobject</td>
<td>Dbobject</td>
</tr>
<tr>
<td>T</td>
<td>Template</td>
</tr>
<tr>
<td>S</td>
<td>Schema</td>
</tr>
</tbody>
</table>

4. On the *Setting* tab, check all the boxes.

5. Save the configuration.

6. In *Control Panel*, click *Administrative Tools*, and then click *Internet Services Manager*. 
7. Add a virtual root to the Web site name "Demodir", and configure it to refer to the same physical directory where you installed the demo files.

8. Save the configuration. Do not edit the "Demos" virtual directory entry. You should now have access to the files in the Northwind database.

To install the stored procedure and table:

1. Open SQL Query Analyzer.

2. Connect to the Northwind database.

3. Open the Mycustomers.sql.

4. Execute the query.

5. Open Mysproc.sql.

6. Execute the query.

7. Leave Query Manager running so you can run the OpenXML demos.

You can now run Url.htm.

See Also

XML Samples
Samples
**XMLStartup**

Microsoft® SQL Server™ 2000 introduces various new features to support XML functionality. The combination of these features makes SQL Server an XML-enabled database server.

This example shows how to:

- Configure Microsoft Internet Information Services (IIS) and SQL Server virtual domains.
- Execute queries and stored procedures from a URL.
- Execute template files.

There are four exercises in this example:

<table>
<thead>
<tr>
<th>Exercise</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exercise1</td>
<td>Configure URL access to the <strong>Northwind</strong> database on a local instance of SQL Server. Your local server is also running IIS and will act as its own host.</td>
</tr>
<tr>
<td>Exercise2</td>
<td>Execute different types of queries against the <strong>Northwind</strong> database.</td>
</tr>
<tr>
<td>Exercise3</td>
<td>Execute stored procedures using a browser.</td>
</tr>
<tr>
<td>Exercise4</td>
<td>Execute queries against the <strong>Northwind</strong> database using template files.</td>
</tr>
</tbody>
</table>

For more details about how to use this example, please refer to the file, C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Xml\Xmlstartup\Xmlstartup.doc.

**Default Location**

C:\Program Files\Microsoft SQL
See Also

XML Samples
Samples
SQL-SCM Samples

The following samples illustrate Microsoft® SQL Server™ 2000 SQL-SCM development.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCMControl</td>
<td>C++ Sample. Illustrates using SQL Service Control Manager (SQL-SCM) to query and change the execution state of services on a local computer.</td>
</tr>
<tr>
<td>SCMStatus</td>
<td>C++ Sample. Illustrates using SQL Service Control Manager (SQL-SCM) w95scm.dll to query the execution state of services on a local computer.</td>
</tr>
</tbody>
</table>

To install the samples during SQL Server installation

1. On the Setup Type page, select Custom.

2. On the Select Components page, under Components, select Code Samples.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_ado.misc, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc.

Prerequisites

C and C++ samples require Microsoft Visual C++ version 6.0.

See Also

Samples
Samples
SCMControl

This sample illustrates using SQL Service Control Manager (SQL-SCM) to query and change the execution state of the following services on a local computer:

- Microsoft® SQL Server (MSSQLServer)
- SQL Server Agent (SQLServerAgent)
- MS DTC (MSDTC)

SQL Service Control Manager (SQL-SCM) is an API supporting control of the Microsoft® SQL Server™ and related services for Microsoft Windows® 95/98 and Microsoft Windows NT®.

This Microsoft® Visual C++® sample uses the w95scm.dll to get the status of and control various SQLServer services. The sample is a console application. Using this sample, you can:

- Start the SQL Server (MSSQLServer), SQL Server Agent (SQLServerAgent) or MS DTC (MSDTC) services on the local computer.
- Pause and continue the MSSQLServer service.
- Stop a supported service.
- Query a supported service to determine its current execution state.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc\Scm\SCMControl
Running the Sample

To run this program, open SCM.dsw workspace and compile the program. Make sure the w95scm.dll is in the execution path.

See Also

SQL-SCM Samples
Samples
SCMStatus

This sample illustrates using SQL Service Control Manager (SQL-SCM) w95scm.dll to query the execution state of the following services on a local computer:

- Microsoft® SQL Server (MSSQLServer)
- SQL Server Agent (SQLServerAgent)
- MS DTC (MSDTC)

SQL Service Control Manager (SQL-SCM) is an API supporting control of the Microsoft® SQL Server™ and related services for Microsoft Windows® 95/98 and Microsoft Windows NT®. This Microsoft® Visual C++® sample is a console application.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Misc\Scm\SCMStatus

Running the Sample

To run this program, open the SCM.dsw workspace and compile the program. Make sure the w95scm.dll is in the execution path.

See Also

SQL-SCM Samples
Samples
MS DTC Samples

The following samples illustrate using Microsoft® SQL Server™ 2000 with Microsoft Distributed Transaction Coordinator (MS DTC), using various technologies.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS DTC Dblib Sample</td>
<td>This program demonstrates how to use MS DTC in C++ using DB-Library.</td>
</tr>
<tr>
<td>MS DTC ODBC Sample</td>
<td>This program demonstrates how to use MS DTC in C++ using ODBC.</td>
</tr>
<tr>
<td>MS DTC T-SQL Sample</td>
<td>This sample demonstrates how to use MS DTC in a Transact-SQL script.</td>
</tr>
</tbody>
</table>

To install the samples during SQL Server installation

1. On the Setup Type page, select Custom.

2. On the Select Components page, under Components, select Code Samples.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_msdtc.exe, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Msdtc.

Prerequisites

C samples require Microsoft Visual C++ version 6.0.

See Also

Samples
Samples
MS DTC Dblib Sample

This program demonstrates how to use MS DTC in C++ using DB-Library.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Msdtc\Dblib

Running the Sample

Open dlib.dsw in Microsoft® Visual C++® 6.0 and compile this program. Make sure to do the following before compiling:

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included and appear at the top of the list:
   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where sqlcurs.exe is located. Run the program by typing "dlib -h".

See Also

MS DTC Samples
Samples
MS DTC ODBC Sample

This program demonstrates how to use MS DTC in C++ using ODBC. It uses MS DTC to perform simultaneous updates on two SQL servers. The transaction in this example is client initiated. The client also initiates the commit operation.

Because this sample uses the ODBC interface, you'll need to configure ODBC data source names for two instances of SQL server to run this sample. Each data source name (DSN) must reference the pubs database in a separate instance of SQL Server. The authors table in the pubs database is used in this sample.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Msdtc\Odbc

Running the Sample

Open odbc.dsw in Microsoft® Visual C++® 6.0 and compile this program. Make sure to do the following before compiling:

1. From the Tools menu, choose Options, and then click the Directories tab.

2. From the Show directories for box, choose Include files and Library files, and ensure that these directories (as appropriate) are included and appear at the top of the list:
   - Include files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Include
   - Library files: C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Lib

To run the sample, open a Command Prompt window, and then change the current directory to where sqlcurs.exe is located. Run the program by entering "odbc -h".
See Also

MS DTC Samples
Samples
MS DTC T-SQL Sample

This sample demonstrates how to use MS DTC in a Transact-SQL script.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Msdtc\Tsql

Running the Sample

1. Run Updadr.sql to create a stored procedure.

2. Run Chgaddr.sql to run the stored procedure using MS DTC.

See Also

MS DTC Samples
Samples
OLE Automation Samples

The following samples illustrate Microsoft® SQL Server™ 2000 OLE Automation application development.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Getnpv OLE Automation Sample</td>
<td>Microsoft Visual Basic® sample. Returns a net present value from a query.</td>
</tr>
<tr>
<td>Loopback (DAO) OLE Automation Sample</td>
<td>Visual Basic sample. Retrieves contents of the input table name provided to the DAO table and returns a tabular array equivalent to the result set returned from SQL Server.</td>
</tr>
<tr>
<td>Loopback (RDO) OLE Automation Sample</td>
<td>Visual Basic sample. Uses RDO to call SQL Server and get the contents of the authors table, and returns a tabular array equivalent to the result set returned from SQL Server.</td>
</tr>
<tr>
<td>Traverse OLE Automation Sample</td>
<td>Transact-SQL sample. Demonstrates how to use traversal syntax for object hierarchies.</td>
</tr>
</tbody>
</table>

To install the samples during SQL Server installation

1. On the Setup Type page, select Custom.

2. On the Select Components page, under Components, select Code Samples.

Samples are installed as a self-extracting file. To extract the samples, double-click Unzip_oleauto.exe, located at C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Oleauto.

Prerequisites

Visual Basic samples require Microsoft Visual Basic version 6.0.

See Also
Samples
Samples
Getnpv OLE Automation Sample

This Microsoft® Visual Basic® sample returns a net present value from a query.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Oleauto\Getnpv

See Also

OLE Automation Samples
Samples
Loopback (DAO) OLE Automation Sample

This Microsoft® Visual Basic® sample retrieves contents of the input table name provided to the DAO table, and returns a tabular array equivalent to the result set returned from Microsoft SQL Server™.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Oleauto\Loopback.dao

See Also

OLE Automation Samples
Samples
Loopback (RDO) OLE Automation Sample

This Microsoft® Visual Basic® sample uses RDO to call Microsoft SQL Server™ and get the contents of the authors table, and returns a tabular array equivalent to the result set returned from SQL Server.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Oleauto\Loopback.rdo

See Also

OLE Automation Samples
Samples
Traverse OLE Automation Sample

This sample is a Transact-SQL script that demonstrates how to use traversal syntax for object hierarchies.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Oleauto\Traverse

See Also

OLE Automation Samples
Samples
Database Schema Samples


You can review these samples to learn about database schema construction, use them as a basis for building your own custom schemas, or gain hands-on experience by working with the schemas while reading *The Data Model Resource Book*. For detailed descriptions and ideas for deploying the Silverston database schemas in database and modeling tools, you should refer to *The Data Model Resource Book*.

The Silverston database schema samples are available only with SQL Server 2000 Enterprise Edition.

Each sample consists of SQL scripts that populate a database with tables and columns, and a Microsoft Word document that describes schema definitions. There is no sample data for the sample schemas. The following samples are included:

- **Sample Corporate Data Model**
- **Sample Enterprise Data Warehouse**
- **Sample Human Resources Data Mart**
- **Sample Sales Data Mart**

Installing Samples

To install the samples, you can run a custom setup and select **Code Samples**. You can also select **Code Samples** when you run Setup to add components to your existing installation.

Samples are installed as a self-extracting file. To extract the samples, double-
click Unzip_silverstondb.exe, located at C:\Program Files\Microsoft SQL Server\80\Devtools\Samples\silverstondb.

**See Also**

[Code Samples](#)

[Setup Type: Typical, Minimum, or Custom](#)
Samples
Sample Corporate Data Model

The corporate data model is a sample database schema published by Silverston, Inmon, and Graziano, and it is licensed for use with Microsoft® SQL Server™ 2000. This sample includes definitions of tables frequently used when implementing models of an enterprise.

For this sample, an additional file called Tabdsc.txt contains detailed table descriptions for the corporate data model.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Silverstondb\Common

Running the Samples

1. In SQL Server Enterprise Manager, create a database to store the sample schema elements.

2. In SQL Query Analyzer, run Common_tables.sql to create the tables, and then run Common_constraints.sql to create primary and foreign key constraints.

3. In SQL Server Enterprise Manager, open the database you just created to view the contents.

See Also

Database Schema Samples
Samples
Sample Enterprise Data Warehouse

The enterprise data warehouse is a sample database schema published by Silverston, Inmon, and Graziano, and licensed for use with Microsoft® SQL Server™ 2000. This sample includes definitions of tables frequently used when implementing models of an enterprise data warehouse.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Silverstondb\DWEnt

Running the Samples

1. In SQL Server Enterprise Manager, create a database to store the sample schema elements.

2. In SQL Query Analyzer, run Dwent_tables.sql to create the tables, and then run Dwent_constraints.sql to create primary and foreign key constraints.

3. In SQL Server Enterprise Manager, open the database you just created to view the contents.

See Also

Database Schema Samples
Samples
Sample Human Resources Data Mart

The human resources data mart is a sample database schema published by Silverston, Inmon, and Graziano, and licensed for use with Microsoft® SQL Server™ 2000. This sample includes definitions of tables frequently used when implementing models of a Human Resources department.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Silverstondb\DWHr

Running the Samples

1. In SQL Server Enterprise Manager, create a database to store the sample schema elements.

2. In SQL Query Analyzer, run Dwhr_tables.sql to create the tables, and then run Dwhr_constraints.sql to create primary and foreign key constraints.

3. In SQL Server Enterprise Manager, open the database you just created to view the contents.

See Also

Database Schema Samples
Samples
Sample Sales Data Mart

The sales data mart is a sample database schema published by Silverston, Inmon, and Graziano, and licensed for use with Microsoft® SQL Server™ 2000. This sample includes definitions of tables frequently used when implementing models of Sales department.

Default Location

C:\Program Files\Microsoft SQL Server\80\Tools\Devtools\Samples\Silverstondb\DWSales

Running the Samples

1. In SQL Server Enterprise Manager, create a database to store the sample schema elements.

2. In SQL Query Analyzer, run Dwsales_tables.sql to create the tables, and then run Dwsales_constraints.sql to create primary and foreign key constraints.

3. In SQL Server Enterprise Manager, open the database you just created to view the contents.

See Also

Database Schema Samples