# **Getting Started with Command Prompt Utilities**

The command prompt utilities are installed automatically when you install the Microsoft® SQL Server<sup>™</sup> 2000 utilities on a computer running Microsoft Windows® 2000, Microsoft Windows NT®, Microsoft Windows 95, or Microsoft Windows 98. The table shows the utilities and the directories where they are installed.

Directory	Utilities
x:\Program Files\Microsoft SQL Server\MSSQL\Binn <sup>1</sup>	bcp <sup>2</sup> console isql <sup>2</sup> sqlagent sqldiag sqlmaint sqlservr vswitch
x:\Program Files\Microsoft SQL Server\80\Tools\Binn	bcp <sup>2</sup> dtsrun dtswiz isql <sup>2</sup> isqlw itwiz odbccmpt osql rebuildm salftwiz
<i>x</i> :\Program Files\Microsoft SQL Server\80\Com	distrib <sup>3</sup> logread <sup>3</sup> replmerg <sup>3</sup> snapshot <sup>3</sup>
<i>x</i> :\Program Files\Common Files\Microsoft Shared\Service Manager	scm
x:\Program Files\Common Files\Microsoft Shared\???	regxmlss

1 MSSQL is the directory name for the default instance of SQL Server 2000. For each named instance of SQL Server 2000, the corresponding directory name is MSSQL\$*instance\_name*. 2 The **bcp** and **isql** utilities are installed in both the \MSSQL\Binn directory and the \80\Tools\Binn directory. They are also installed in the \MSSQL\$*instance\_name*\Binn directory for each named instance of SQL Server 2000. Any copy of these utilities may be used to connect to any instance of SQL Server. The instance of SQL Server to connect to is determined for each utility by a server argument. You are not limited to the instance of SQL Server corresponding to the directory in which the utility is run. 3 These are the file names for the four replication agent utilities.

During installation, the *x*:\Program Files\Microsoft SQL Server\80\Tools\Binn directory is added to the system path. You can run the utilities in this directory at any command prompt. For a utility not in the 80\Tools\Binn directory, you must either run the utility from a command prompt in the directory in which it is installed or explicitly specify the path.

These utilities are no longer installed by SQL Server 2000 Setup.

- makepipe utility
- **odbcping** utility
- **readpipe** utility

If you need to run these utilities, you can run them from the *x*:\x86\Binn directory on the SQL Server 2000 compact disc, or manually copy them to your computer.

# **Command Prompt Utilities Syntax Conventions**

Convention	Used for
UPPERCASE	Statements and terms used at the operating system level.
monospace	Sample commands and program code.
italic	User-supplied parameters.
bold	Commands, parameters, and other syntax that must be typed exactly as shown.

# **bcp** Utility

The **bcp** utility copies data between an instance of Microsoft® SQL Server<sup>™</sup> 2000 and a data file in a user-specified format.

# Syntax

bcp {[[database\_name.][owner].]{table\_name | view\_name} | "query"}
 {in | out | queryout | format} data\_file
 [-m max\_errors] [-f format\_file] [-e err\_file]
 [-F first\_row] [-L last\_row] [-b batch\_size]
 [-n] [-c] [-w] [-N] [-V (60 | 65 | 70)] [-6]
 [-q] [-C code\_page] [-t field\_term] [-r row\_term]
 [-i input\_file] [-o output\_file] [-a packet\_size]
 [-S server\_name[\instance\_name]] [-U login\_id] [-P password]
 [-T] [-v] [-R] [-k] [-E] [-h "hint [,...n]"]

# Arguments

database\_name

Is the name of the database in which the specified table or view resides. If not specified, this is the default database for the user.

#### Owner

Is the name of the owner of the table or view. *owner* is optional if the user performing the bulk copy operation owns the specified table or view. If *owner* is not specified and the user performing the bulk copy operation does not own the specified table or view, Microsoft® SQL Server<sup>TM</sup> 2000 returns an error message, and the bulk copy operation is canceled.

#### table\_name

Is the name of the destination table when copying data into SQL Server (**in**), and the source table when copying data from SQL Server (**out**).

#### view\_name

Is the name of the destination view when copying data into SQL Server (in),

and the source view when copying data from SQL Server (**out**). Only views in which all columns refer to the same table can be used as destination views. For more information on the restrictions for copying data into views, see <u>INSERT</u>.

### Query

Is a Transact-SQL query that returns a result set. If the query returns multiple result sets, such as a SELECT statement that specifies a COMPUTE clause, only the first result set is copied to the data file; subsequent result sets are ignored. Use double quotation marks around the query and single quotation marks around anything embedded in the query. **queryout** must also be specified when bulk copying data from a query.

## in | out | queryout | format

Specifies the direction of the bulk copy. **in** copies from a file into the database table or view. **out** copies from the database table or view to a file. **queryout** must be specified only when bulk copying data from a query. **format** creates a format file based on the option specified (**-n**, **-c**, **-w**, **-6**, or **-N**) and the table or view delimiters. If **format** is used, the **-f** option must be specified as well.

**Note** The **bcp** utility included with Microsoft SQL Server 6.5 does not support bulk copying into tables that contain the **sql\_variant** or **bigint** data types.

## data\_file

Is the full path of the data file used when bulk copying a table or view to or from a disk. When bulk copying data into SQL Server, the data file contains the data to be copied into the specified table or view. When bulk copying data from SQL Server, the data file contains the data copied from the table or view. The path can have from 1 through 255 characters.

#### -m max\_errors

Specifies the maximum number of errors that can occur before the bulk copy operation is canceled. Each row that cannot be copied by **bcp** is ignored and counted as one error. If this option is not included, the default is 10.

## -f format\_file

Specifies the full path of the format file that contains stored responses from a previous use of **bcp** on the same table or view. Use this option when using a format file created with the **format** option to bulk copy data in or out. Creation of the format file is optional. After prompting you with format questions, **bcp** prompts whether to save the answers in a format file. The default file name is Bcp.fmt. **bcp** can refer to a format file when bulk copying data; therefore, reentering previous format responses interactively is not necessary. If this option is not used and **-n**, **-c**, **-w**, **-6**, or **-N** is not specified, **bcp** prompts for format information.

#### -e err\_file

Specifies the full path of an error file used to store any rows **bcp** is unable to transfer from the file to the database. Error messages from **bcp** go to the user's workstation. If this option is not used, an error file is not created.

#### -F first\_row

Specifies the number of the first row to bulk copy. The default is 1, indicating the first row in the specified data file.

#### -L last\_row

Specifies the number of the last row to bulk copy. The default is 0, indicating the last row in the specified data file.

#### -b batch\_size

Specifies the number of rows per batch of data copied. Each batch is copied to the server as one transaction. SQL Server commits or rolls back, in the case of failure, the transaction for every batch. By default, all data in the specified data file is copied in one batch. Do not use in conjunction with the **- h** "**ROWS\_PER\_BATCH** = *bb*" option.

#### -n

Performs the bulk copy operation using the native (database) data types of the data. This option does not prompt for each field; it uses the native values.

#### -C

Performs the bulk copy operation using a character data type. This option does not prompt for each field; it uses **char** as the storage type, no prefixes,

 $\$  (tab character) as the field separator, and  $\$  (newline character) as the row terminator.

#### -w

Performs the bulk copy operation using Unicode characters. This option does not prompt for each field; it uses **nchar** as the storage type, no prefixes, \t (tab character) as the field separator, and \n (newline character) as the row terminator. Cannot be used with SQL Server version 6.5 or earlier.

#### -N

Performs the bulk copy operation using the native (database) data types of the data for noncharacter data, and Unicode characters for character data. This option offers a higher performance alternative to the **-w** option, and is intended for transferring data from one SQL Server to another using a data file. It does not prompt for each field. Use this option when you are transferring data that contains ANSI extended characters and you want to take advantage of the performance of native mode. **-N** cannot be used with SQL Server 6.5 or earlier.

#### $\text{-V}\,(60 \mid 65 \mid 70)$

Performs the bulk copy operation using data types from an earlier version of SQL Server. Use this option in conjunction with character (-c) or native (-n) format. This option does not prompt for each field; it uses the default values. For example, to bulk copy date formats supported by the **bcp** utility provided with SQL Server 6.5 (but no longer supported by ODBC) into SQL Server 2000, use the **-V 65** parameter.

**IMPORTANT** When bulk copying data from SQL Server into a data file, the **bcp** utility does not generate SQL Server 6.0 or SQL Server 6.5 date formats for any **datetime** or **smalldatetime** data, even if **-V** is specified. Dates are always written in ODBC format. Additionally, null values in **bit** columns are written as the value 0 because SQL Server versions 6.5 and earlier do not support nullable **bit** data.

#### -6

Performs the bulk copy operation using SQL Server 6.0 or SQL Server 6.5 data types. Supported for backward compatibility only. Use the **-V** option

instead.

#### -q

Executes the SET QUOTED\_IDENTIFIERS ON statement in the connection between the **bcp** utility and an instance of SQL Server. Use this option to specify a database, owner, table, or view name that contains a space or a quotation mark. Enclose the entire three-part table or view name in double quotation marks (" ").

### -C code\_page

Supported for backward compatibility only. Instead, specify a collation name for each column in the format file or in interactive **bcp**.

Specifies the code page of the data in the data file. *code\_page* is relevant only if the data contains **char**, **varchar**, or **text** columns with character values greater than 127 or less than 32.

Code page value	Description
ACP	ANSI/Microsoft Windows® (ISO 1252).
OEM	Default code page used by the client. This is the default code page used by <b>bcp</b> if <b>-C</b> is not specified.
RAW	No conversion from one code page to another occurs. This is the fastest option because no conversion occurs.
<value></value>	Specific code page number, for example, 850.

#### -t field\_term

Specifies the field terminator. The default is \t (tab character). Use this parameter to override the default field terminator.

#### **-r** row\_term

Specifies the row terminator. The default is \n (newline character). Use this parameter to override the default row terminator.

#### -i input\_file

Specifies the name of a response file, containing the responses to the

command prompt questions for each field when performing a bulk copy using interactive mode (**-n**, **-c**, **-w**, **-6**, or **-N** not specified).

```
-o output_file
```

Specifies the name of a file that receives output from **bcp** redirected from the command prompt.

-a packet\_size

Specifies the number of bytes, per network packet, sent to and from the server. A server configuration option can be set by using SQL Server Enterprise Manager (or the **sp\_configure** system stored procedure). However, the server configuration option can be overridden on an individual basis by using this option. *packet\_size* can be from 4096 to 65535 bytes; the default is 4096.

Increased packet size can enhance performance of bulk copy operations. If a larger packet is requested but cannot be granted, the default is used. The performance statistics generated by **bcp** show the packet size used.

-S server\_name[\instance\_name]

Specifies the instance of SQL Server to connect to. Specify *server\_name* to connect to the default instance of SQL Server on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, **bcp** connects to the default instance of SQL Server on the local computer. This option is required when executing **bcp** from a remote computer on the network.

# -U login\_id

Specifies the login ID used to connect to SQL Server.

# -P password

Specifies the password for the login ID. If this option is not used, **bcp** prompts for a password. If this option is used at the end of the command prompt without a password, bcp uses the default password (NULL).

-T

Specifies that **bcp** connects to SQL Server with a trusted connection, using the security credentials of the network user. *login\_id* and *password* are not

required.

#### -v

Reports the **bcp** utility version number and copyright.

#### -R

Specifies that currency, date, and time data is bulk copied into SQL Server using the regional format defined for the locale setting of the client computer. By default, regional settings are ignored.

#### -k

Specifies that empty columns should retain a null value during the bulk copy operation, rather than have any default values for the columns inserted.

#### -E

Specifies that the values for an identity column are present in the file being imported. If **-E** is not given, the identity values for this column in the data file being imported are ignored, and SQL Server 2000 automatically assigns unique values based on the seed and increment values specified during table creation. If the data file does not contain values for the identity column in the table or view, use a format file to specify that the identity column in the table or view should be skipped when importing data; SQL Server 2000 automatically assigns unique values for the column. For more information, see <u>DBCC CHECKIDENT</u>.

#### **-h** "*hint* [,...*n*]"

Specifies the hint(s) to be used during a bulk copy of data into a table or view. This option cannot be used when bulk copying data into SQL Server 6.*x* or earlier.

Hint	Description
ORDER (column [ASC	Sort order of the data in the data file. Bulk
<b>DESC</b> ] [, <i>n</i> ] <b>)</b>	copy performance is improved if the data
	being loaded is sorted according to the
	clustered index on the table. If the data file
	is sorted in a different order, or there is no
	clustered index on the table, the ORDER

	hint is ignored. The names of the columns supplied must be valid columns in the destination table. By default, <b>bcp</b> assumes the data file is unordered.
<b>ROWS_PER_BATCH</b> = bb	Number of rows of data per batch (as <i>bb</i> ). Used when <b>-b</b> is not specified, resulting in the entire data file being sent to the server as a single transaction. The server optimizes the bulk load according to the value <i>bb</i> . By default, <b>ROWS_PER_BATCH</b> is unknown.
KILOBYTES_PER_BATCH = cc	Approximate number of kilobytes (KB) of data per batch (as <i>cc</i> ). By default, <b>KILOBYTES_PER_BATCH</b> is unknown.
TABLOCK	A table-level lock is acquired for the duration of the bulk copy operation. This hint significantly improves performance because holding a lock only for the duration of the bulk copy operation reduces lock contention on the table. A table can be loaded concurrently by multiple clients if the table has no indexes and <b>TABLOCK</b> is specified. By default, locking behavior is determined by the table option <b>table lock on bulk load</b> .
CHECK_CONSTRAINTS	Any constraints on the destination table are checked during the bulk copy operation. By default, constraints are ignored.
FIRE_TRIGGERS	Specified with the <b>in</b> argument, any insert triggers defined on the destination table will execute during the bulk copy operation. If <b>FIRE_TRIGGERS</b> is not specified, no insert triggers will execute. <b>FIRE_TRIGGERS</b> is ignored for the <b>out</b> ,

# Remarks

Values in the data file being imported for computed or **timestamp** columns are ignored, and SQL Server 2000 automatically assigns values. If the data file does not contain values for the computed or **timestamp** columns in the table, use a format file to specify that the computed or **timestamp** columns in the table should be skipped when importing data; SQL Server automatically assigns values for the column.

Computed and **timestamp** columns are bulk copied from SQL Server to a data file as usual.

SQL Server identifiers, including database names, table or view names, logins, and passwords, can include characters such as embedded spaces and quotation marks. When you specify an identifier or file name at the command prompt that includes a space or quotation mark, enclose the identifier in double quotation marks (" "). Additionally, for owner, table, or view names that contain embedded spaces or quotation marks, you can either specify the **-q** option or enclose the owner, table, or view name in brackets ([ ]) inside of the double quotation marks.

For example, the **Northwind** database has the table **Jane's Orders**, which is owned by user Jane Doe. To bulk copy this table from the **Northwind** database to the Orders.txt file using the login **Jane Doe** and the password **go dba**, execute one of these commands:

bcp "Northwind.Jane Doe.Jane's Orders" out "Jane's Orders.txt" -c -q -

bcp "Northwind.[Jane Doe].[Jane's Orders]" out "Jane's Orders.txt" -c ·

To specify a database name that contains a space or quotation mark, you must use the  $-\mathbf{q}$  option.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## See Also

Copying Data Between Different Collations Using bcp and BULK INSERT Parallel Data Loads SET QUOTED\_IDENTIFIER sp\_tableoption

Using Format Files

# console Utility

The **console** command prompt utility displays backup and restore messages when backing up to or restoring from tape dump devices, and is used by the person responsible for backing up and restoring a database.

# **Syntax**

```
console [/S server_name[\instance_name]] [/P pipe_name]
```

# Arguments

#### /S server\_name[\instance\_name]:

Is the name of the instance of Microsoft® SQL Server<sup>™</sup> 2000 in which to connect. Specify *server\_name* to connect to the default instance of SQL Server on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, the **console** utility connects to the default instance of SQL Server on the local computer. This option is required when executing the **console** utility from a remote computer on the network.

/**P** pipe\_name:

Is the pipe used to start the server.

# Remarks

The **console** utility must be running before a BACKUP or RESTORE statement can proceed for tape dump devices.

Because the **console** workstation (which can be any screen group) displays prompts and messages for the operator, run it in a window on the same Microsoft Windows NT® Workstation running an instance of Microsoft SQL Server.

If two tapes (that is, two backups or restores) are being used at one time, only one console is needed. The program prints messages one at a time, waiting for a response before displaying the next prompt. For this reason, the messages include the name of the physical drive to which they pertain. To stop a backup or restore to a tape, enter **n** in response to any prompt to mount a new tape.

The **console** utility uses Named Pipes. If you want to use **console** on an alternate pipe, you must set up the connection using SQL Server Client Network Utility.

**Note** If the **console** utility has been started and appears to be unusable, wait for one minute and try to use it again (SQL Server 2000 may be in the process of fixing a broken console connection).

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# dtsrun Utility

The **dtsrun** utility executes a package created using Data Transformation Services (DTS). The DTS package can be stored in the Microsoft® SQL Server<sup>™</sup> **msdb** database, a COM-structured storage file, or SQL Server Meta Data Services.

# Syntax

```
dtsrun [/?] |
[
    /[~]S server_name[\instance_name]
    { {/[~]U user_name [/[~]P password]} | /E }
  ]
  {
    {/[~]N package_name }
    | {/[~]G package_guid_string}
    | {/[~]V package_version_guid_string}
  }
  [/[~]M package_password]
  [/[~]F filename]
  [/[~]R repository_database_name]
  [/A global_variable_name:typeid=value]
  [/L log_file_name]
  [/W NT_event_log_completion_status]
  [/Z] [/!X] [/!D] [/!Y] [/!C]
1
```

# Arguments

# /?

Displays the command prompt options.

~

Specifies that the parameter to follow is hexadecimal text representing the

encrypted value of the parameter. Can be used with the /S, /U, /P, /N, /G, /V, /M, /F, and /R options. Using encrypted values increases the security of the command used to execute the DTS package because the server name, password, and so on, are not visible. Use /!Y to determine the encrypted command.

/S server\_name[\instance\_name]

Specifies the instance of SQL Server to connect to. Specify *server\_name* to connect to the default instance of SQL Server on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server.

/U user\_name

Is a login ID used to connect to an instance of SQL Server.

/P password

Is a user-specified password used with a login ID.

/E

Specifies a trusted connection (password not required).

/N package\_name

Is the name of a DTS package assigned when the package was created.

/G package\_guid\_string

Is the package ID assigned to the DTS package when it was created. The package ID is a GUID.

/V package\_version\_guid\_string

Is the version ID assigned to the DTS package when it was first saved or executed. A new version ID is assigned to the DTS package each time it is modified. The version ID is a GUID.

/M package\_password

Is an optional password assigned to the DTS package when it was created.

## /**F** filename

Is the name of a structured storage file containing DTS packages. If *server\_name* is also specified, the DTS package retrieved from SQL Server is executed and that package is added to the structured storage engine.

#### /R repository\_database\_name

Is the name of the repository database containing DTS packages. If no name is specified, the default database name is used.

### /A global\_variable\_name:typeid=value

Specifies a package global variable, where *typeid* = type identifier for the data type of the global variable. The entire argument string can be quoted. This argument can be repeated to specify multiple global variables. See the Remarks section for the different available type identifiers available with global variables.

To set global variables with this command switch, you must have either Owner permission for the package or the package must have been saved without DTS password protection enabled. If you do not have Owner permission, you can specify global variables, but the values used will be those set in the package, not those specified with the /A command switch.

#### /L log\_file\_name:

Specifies the name of the package log file.

#### /W Windows\_Event\_Log

Specifies whether or not to write the completion status of the package execution to the Windows Application Log. Specify **True** or **False**.

#### /**Z**

Indicates that the command line for **dtsrun** is encrypted using SQL Server 2000 encryption.

#### /!X

Blocks execution of the selected DTS package. Use this command parameter when you want to create an encrypted command line without executing the DTS package.

#### /!D

Deletes the DTS package from an instance of SQL Server. The package is not executed. It is not possible to delete a specific DTS package from a structured storage file. The entire file needs to be overwritten using the /**F** and /**S** options.

## /!Y

Displays the encrypted command used to execute the DTS package without executing it.

### /!C

Copies the command used to execute the DTS package to the Microsoft Windows® clipboard. This option can also be used in conjunction with **/!X** and **/!Y**.

# Remarks

If you do not specify any command line switches, specify an incorrect command line switch, or your command statement contains a syntax error, **dtsrun** returns error information and usage instructions. If you enter **dtsrunui** on the command line without any command line switches, you will start the DTS Run utility.

Spaces between command switches and values are optional. Embedded spaces in values must be embedded between double quotation marks.

If an option is specified multiple times, the last occurrence takes precedence. One exception is the /A command switch. Specifying more than one /A switch creates multiple global variables.

When specifying a global variable with the /**A** command switch, you must use a type identifier to indicate the data type of the global variable.

A tilde (~) character after the forward slash (for example, /~Z) indicates that the parameter value is encrypted and what follows is the hexadecimal text of the encrypted value.

The table shows the global variable data types and their IDs.

Data type	Type ID
Integer (small)	2
Integer	3

Real (4-byte)	4
Real (8-byte)	5
Currency	6
Date	7
String	8
Boolean	11
Decimal	14
Integer (1-byte)	16
Unsigned int (1-byte)	17
Unsigned int (2-byte)	18
Unsigned int (4-byte)	19
Integer (8-byte)	20
Unsigned int (8-byte)	21
Int	22
Unsigned int	23
HRESULT	25
Pointer	26
LPSTR	30
LPWSTR	31

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# Examples

To execute a DTS package saved as a COM-structured storage file, use:

dtsrun /Ffilename /Npackage\_name /Mpackage\_password

To execute a DTS package saved in the SQL Server **msdb** database, use:

dtsrun /Sserver\_name /Uuser\_nName /Ppassword /Npackage\_name /N

To execute a DTS package saved in Meta Data Services, use:

dtsrun /Sserver\_name /Uuser\_nrame /Ppassword /Npackage\_name /M

# See Also

Executing a DTS Package

# dtswiz Utility

The **dtswiz** utility starts the DTS Import/Export Wizard using command prompt options. The wizards can be used to create Data Transformation Services (DTS) packages that import, export, or transform data between data sources, for example between an instance of Microsoft® SQL Server<sup>™</sup> 2000 and a Microsoft Access database, ASCII text file, or any ODBC data source.

# Syntax

```
dtswiz [/?] |
[
[
        [ /i | /x}
        [/r provider_name]
        [/s server_name[\instance_name]]
        [/n | /u login_id [/p password]]
        [/d database_name]
    ]
    [/f filename]
    [/y] [/m]
]
```

# Arguments

# /?

Displays the command prompt options.

# /i

Specifies an import to an instance of SQL Server.

#### /**x**

Specifies an export from an instance of SQL Server.

#### /**r** provider\_name:

Is the name of the provider used to connect to the data source when

importing, or the destination when exporting. For example, the Microsoft OLE DB Provider for ODBC is MSDASQL.

```
/s server_name[\instance_name]
```

Is the instance of SQL Server to export data from or import data to. Specify *server\_name* to connect to the default instance of SQL Server on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server.

### /n

Specifies Windows Authentication (not required). If used, /**n** takes precedence over /**u** and /**p**.

#### /**u** login\_id

Is a login ID used to connect to an instance of SQL Server.

#### /p password

Is a user-specified password used with a login ID.

#### /**d** database\_name

Is the SQL Server database used to export data from, or import data to.

#### /**f** filename

Saves the DTS package created by the wizard to this COM-structured storage file.

## /y

Hides the SQL Server system databases (**master**, **model**, **msdb**, **tempdb**). These databases do not show up in the list of source databases when importing data, or the list of destination databases when exporting data.

#### /m

Execute all steps on the main package thread. For more information, see <u>Enhancing Performance of DTS Packages</u>.

# Remarks

Any number of these options can be used as parameters to the **dtswiz** command. The DTS Import/Export Wizard request any required values not supplied.

The DTS package created can be saved to the SQL Server **msdb** database, a COM-structured storage file, or Microsoft SQL Server 2000 Meta Data Services.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## See Also

Creating a DTS Package with the DTS Import/Export Wizard

# isql Utility

The **isql** utility allows you to enter Transact-SQL statements, system procedures, and script files; and uses DB-Library to communicate with Microsoft® SQL Server<sup>™</sup> 2000.

# Syntax

```
isql
  [-?]|
  [-L]|
  [
     {
       {-U login_id [-P password]}
       |-E
     }
    [-S server_name] [-H wksta_name] [-d db_name]
    [-l time_out] [-t time_out] [-h headers]
    [-s col_separator] [-w column_width] [-a packet_size]
    [-e] [-x max_text_size]
    [-c cmd_end] [-q "query"] [-Q "query"]
    [-n] [-m error_level] [-r {0 | 1}]
    [-i input_file] [-o output_file] [-p]
    [-b][-O]
  ]
```

# Arguments

## -?

Displays the syntax summary of **isql** switches.

#### -L

Lists the locally configured servers and the names of the servers broadcasting on the network.

-U login\_id

Is the user login ID. Login IDs are case-sensitive.

-P password

Is a user-specified password. If the **-P** option is not used, **isql** prompts for a password. If the **-P** option is used at the end of the command prompt without any password, **isql** uses the default password (NULL). Passwords are case-sensitive.

The ISQLPASSWORD environment variable allows you to set a default password for the current session. Therefore, you do not have to hard code a password into batch files.

If you do not specify a password with the **-P** option, **isql** first checks for the ISQLPASSWORD variable. If no value is set, **isql** uses the default password, NULL. The following example sets the ISQLPASSWORD variable at the command prompt and then accesses the **isql** utility:

# C:\>SET ISQLPASSWORD=abracadabra C:\>isql

-E

Uses a trusted connection instead of requesting a password.

-S server\_name

Specifies the default instance of SQL Server to connect to. **isql** does not support connecting to a named instance of SQL Server 2000. If no server is specified, **isql** connects to the default instance of SQL Server on the local computer. This option is required if you are executing **isql** from a remote computer.

-H wksta\_name

Is a workstation name. The workstation name is stored in **sysprocesses.hostname** and is displayed by **sp\_who**. If not specified, the current computer name is assumed.

#### **-d** *db\_name*

Issues a USE *db\_name* statement when **isql** is started.

#### -l time\_out

Specifies the number of seconds before an **isql** login times out. If no *time\_out* value is specified, a command runs indefinitely. The default time-out for login to **isql** is eight seconds.

#### **-t** time\_out

Specifies the number of seconds before a command times out. If no *time\_out* value is specified, a command runs indefinitely; the default time-out for logging in to **isql** is eight seconds.

#### -h headers

Specifies the number of rows to print between column headings. The default is to print headings one time for each set of query results. Use **-1** to specify that no headers will be printed. If using **-1**, there must be no space between the parameter and the setting (**-h-1**, not **-h -1**).

#### -s col\_separator

Specifies the column-separator character, which is a blank space by default. To use characters that have special meaning to the operating system (for example, |; & < >), enclose the character in double quotation marks (").

#### -w column\_width

Allows the user to set the screen width for output. The default is 80 characters. When an output line has reached its maximum screen width, it is broken into multiple lines.

#### -a packet\_size

Allows you to request a different-sized packet. The valid values for *packet\_size* are 512 through 65535. The default value for the Microsoft Windows NT® version of **isql** is 8192; otherwise, the default value is 512 for Microsoft MS-DOS®, although larger sizes can be requested with that version as well. Increased packet size can enhance performance on larger script execution where the amount of SQL statements between GO commands is substantial. Microsoft testing indicates that 8192 is typically the fastest setting for bulk copy operations. A larger packet size can be requested, but **isql** defaults to 512 if the request cannot be granted.

Echoes input.

```
-x max_text_size
```

Specifies, in bytes, the maximum length of text data to return. Text values longer than *max\_text\_size* are truncated. If *max\_text\_size* is not specified, text data is truncated at 4096 bytes.

#### -c cmd\_end

Specifies the command terminator. By default, commands are terminated and sent to SQL Server 2000 by entering GO on a line by itself. When you reset the command terminator, do not use Transact-SQL reserved words or characters that have special meaning to the operating system, whether preceded by a backslash or not.

## -q "query"

Executes a query when **isql** starts, but does not exit **isql** when the query completes. (Note that the query statement should not include GO). If you issue a query from a batch file, you can use %variables. Environment %variables% also work. For example:

SET table = sysobjects isql /q "Select \* from %table%"

Use double quotation marks around the query and single quotation marks around anything embedded in the query.

-Q "query"

Executes a query and immediately exits **isql** when the query completes. Use double quotation marks around the query and single quotation marks around anything embedded in the query.

#### -n

Removes numbering and the prompt symbol (>) from input lines.

-m error\_level

Customizes the display of error messages. The message number, state, and

-e

error level are displayed for errors of the specified severity level or higher. Nothing is displayed for errors of severity levels lower than the specified level. Use **-1** to specify that all headers are returned with messages, even informational messages. If **-1** is used, there must be no space between the parameter and the setting (**-m-1**, not **-m -1**).

-r  $\{0 \mid 1\}$ 

Redirects message output to the screen (**stderr**). If you do not specify a parameter, or if you specify **0**, only error messages with severity 17 or higher are redirected. If you specify **1**, all message output (including "print") is redirected.

-i input\_file

Identifies the file that contains a batch of SQL statements or stored procedures. The less than (<) comparison operator can be used in place of **-i**.

-o output\_file

Identifies the file that receives output from **isql**. The greater than (>) comparison operator can be used in place of **-o**.

# -p

Prints performance statistics.

## -b

Specifies that **isql** exits and returns a DOS ERRORLEVEL value when an error occurs. The value returned to the DOS ERRORLEVEL variable is 1 when the SQL Server error message has a severity of 10 or greater; otherwise, the value returned is 0. MS-DOS batch files can test the value of DOS ERRORLEVEL and handle the error appropriately.

## -0

Specifies that **isql** reverts to the behavior of earlier versions. These features are deactivated:

- EOF batch processing
- Automatic console width scaling
• Wide messages

This option also sets the default DOS ERRORLEVEL value to -1.

## Remarks

All DB-Library applications, such as **isql**, work as SQL Server 6.5–level clients when connected to SQL Server 2000. They do not support some SQL Server 2000 features. The **osql** utility is based on ODBC and does support all SQL Server 2000 features. Use **osql** to run scripts that **isql** cannot run. For more information about the restrictions on SQL Server 6.5–level clients, see <u>Connecting Early Version Clients to SQL Server 2000</u> in SQL Server Books Online.

The SQL Query Analyzer default is to save SQL scripts as Unicode files. The **isql** utility does not support Unicode input files. Attempting to specify one of these files in the **-i** switch results in a 170 error:

Incorrect syntax near ' '.

Use the **osql** utility to run these Unicode files. An alternative is to specify ANSI instead of Unicode in the **File format** list of the SQL Query Analyzer **File/Save As** dialog box.

Like most DB-Library applications, the **isql** utility does not set any connection options by default. Users must issue SET statements interactively or in their scripts if they want to use specific connection option settings.

The **isql** utility is started directly from the operating system with the casesensitive options listed here. After starting, **isql** accepts Transact-SQL statements and sends them to SQL Server 2000 interactively. The results are formatted and printed on the standard output device (the screen). Use QUIT or EXIT to exit from **isql**.

If you do not specify a user name when you start **isql**, SQL Server 2000 checks for the environment variables and uses those, for example, **isqluser=(***user***)** or **isqlserver=(***server***)**. If no environment variables are set, the workstation user name is used. If you do not specify a server, the name of the workstation is used.

If neither the **-U** or **-P** options are used, SQL Server 2000 attempts to connect using Windows Authentication Mode. Authentication is based on the Windows NT account of the user running **isql**.

In addition to using Transact-SQL statements within **isql**, the commands shown in this table are also available.

Command	Description
GO	Executes all statements entered after the last GO.
RESET	Clears any statements you have entered.
ED	Calls the editor.
<b>!!</b> command	Executes an operating-system command.
QUIT or EXIT()	Exits from <b>isql</b> .
CTRL+C	Ends a query without exiting from <b>isql</b> .

The command terminators GO (by default), RESET, ED, !!, EXIT, QUIT, and CTRL+C are recognized only if they appear at the beginning of a line, immediately following the **isql** prompt. Anything entered on the same line after these keywords is disregarded by **isql**.

GO signals both the end of a batch and the execution of any cached Transact-SQL statements. When you press ENTER at the end of each input line, **isql** caches the statements on that line. When you press ENTER after typing GO, all of the currently cached statements are sent as a batch to SQL Server 2000.

The current **isql** utility works as if there is an implied GO at the end of any script executed, therefore all statements in the script execute. Some earlier versions of **isql** do not send any statements to the server unless there is at least one GO in an input script. Any statements after the last GO are not executed.

End a command by typing a line beginning with a command terminator. You can follow the command terminator with an integer to specify how many times the command should be run. For example, to execute this command 100 times, type:

SELECT x = 1 GO 100

The results are printed once, at the end of execution. With **isql**, there is a limit of

1000 characters per line. Large statements should be spread across multiple lines.

The user can call an editor on the current query buffer by typing ED as the first word on a line. The editor is defined in the EDITOR environment variable. The default editor is "edit" for MS-DOS and Windows NT. You can specify a different editor by setting the EDITOR environment variable. For example, to make the default editor Notepad, enter at the operating-system prompt:

## SET EDITOR=notepad

For more information about where to find or how to run this utility, see <u>Getting</u> <u>Started with Command Prompt Utilities</u>.

# **Operating-System Commands**

Operating-system commands can also be executed by starting a line with two exclamation points (!!) followed by the command. The command recall facilities of DOSKEY can be used to recall and modify previously entered **isql** statements on a computer running Windows NT. The existing query buffer can be cleared by typing RESET.

When running stored procedures, **isql** prints a blank line between each set of results in a batch. In addition, the "0 rows affected" message does not appear when it does not apply to the statement executed.

# Using isql Interactively

To use **isql** interactively, type the **isql** command (and any of the options) at a command prompt.

You can read in a file containing a query (such as Stores.qry) for execution by **isql** by typing a command similar to this:

```
isql /U alma /P /i stores.qry
```

The file must include a command terminator(s).

You can read in a file containing a query (such as Titles.qry) and direct the results to another file by typing a command similar to this:

## isql /U alma /P /i titles.qry /o titles.res

When using **isql** interactively, you can read an operating-system file into the command buffer with **:r** *file\_name*. Do not include a command terminator in the file; enter the terminator interactively after you have finished editing.

## **Inserting Comments**

You can include comments in a Transact-SQL statement submitted to SQL Server 2000 by **isql**. Two types of commenting styles are allowed: -- and /\*...\*/.

For more information, see <u>Using Comments</u>.

# Using EXIT to Return Results in isql

You can use the result of a SELECT statement as the return value from **isql**. The first column of the first result row is converted to a 4-byte integer (long). MS-DOS passes the low byte to the parent process or operating-system error level. Windows NT passes the entire 4-byte integer. The syntax is:

EXIT(query)

For example:

```
EXIT(SELECT @@rowcount)
```

```
EXIT(SELECT 5)
```

You can also include the EXIT parameter as part of a batch file. For example:

```
isql /Q "EXIT(SELECT COUNT(*) FROM '%1')"
```

The **isql** utility passes everything between the parentheses () to the server exactly as entered. The EXIT() statement can span lines. If a stored system procedure selects a set and returns a value, only the selection is returned. The EXIT() statement with nothing between the parentheses executes everything preceding it in the batch and then exits with no return value.

There are four EXIT formats:

• EXIT

Does not execute the batch; quits immediately and returns no value.

• EXIT()

Executes the batch, and then quits and returns no value.

• EXIT(query)

Executes the batch, including the query, and then quits after returning the results of the query.

• RAISERROR with a state of 127.

If RAISERROR is used within an **isql** script and a state of 127 is raised, **isql** will quit and return the message ID back to the client. For example:

RAISERROR(50001, 10, 127)

This error will cause the **isql** script to end and the message ID 50001 will be returned to the client.

The return values -1 through -99 are reserved by SQL Server; **isql** defines the following values:

• -100

Error encountered prior to selecting return value.

• -101

No rows found when selecting return value.

• -102

Conversion error when selecting return value.

# See Also

/\*...\*/ (Comment) -- (Comment) Managing Security

## RAISERROR

Command Prompt Utilities

# isqlw Utility

The **isqlw** utility (SQL Query Analyzer) allows you to enter Transact-SQL statements, system stored procedures, and script files. You can set up shortcuts or create batch files to launch a preconfigured SQL Query Analyzer.

# Syntax

```
isqlw [-?] |
[
[-S server_name[\instance_name]]
[-d database]
[-E] [-U user] [-P password]
[{-i input_file} {-o output_file} [-F {U|A|O}]]
[-f file_list]
[-C configuration_file]
[-D scripts_directory]
[-T template_directory]
]
```

## Arguments

-?

Displays usage information.

## -S server\_name[\instance\_name]:

Specifies the instance of Microsoft® SQL Server<sup>™</sup> 2000 to connect to. Specify *server\_name* to connect to the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, **isqlw** connects to the default instance of SQL Server on the local computer. This option is required when executing **isqlw** from a remote computer on the network.

## -d database

Issues a USE *database* statement when **isqlw** is started. The default is the

default database of the user.

-E

Uses a trusted connection instead of requesting a password.

-U user

Is the user login ID. Login IDs are case-sensitive.

-P password

Is the login password. The default is NULL.

-i input\_file

Identifies the file that contains a batch of SQL statements or stored procedures. The **–i** and **–o** options must be specified together. When **-i** and **-o** options are specified, queries in the input file are executed and results are saved in the output file. No user interface is shown while the queries are executing. When execution is completed, the process exits.

-o output\_file

Identifies the file that receives output from **isqlw**. The **–i** and **–o** options must be specified together. When **-i** and **-o** options are specified, queries in the input file are executed and results are saved in the output file. No user interface is shown while the queries are executing. When execution is completed, the process exits. If file format is not specified with **-F**, the output file will be of the same type as the input file.

-F  $\{U|A|O\}$ 

Is the format of the input and output files. Values include Unicode, ANSI, and OEM. If **-F** is not specified, automatic mode is used (if the file is Unicode signed, the file is opened as Unicode; otherwise, the file is opened as ANSI).

## -f file\_list

Loads the listed files into SQL Query Analyzer. With the **-f** option, you can load one or more files (file names separated by single space). If more than one file is specified, files are opened on the same connection context. The file name can include the directory path where the file resides. Wildcard

characters, such as the asterisk (\*), as in C:\Test\\*.sql, can be used.

-C configuration\_file

Uses the settings specified in the configuration file. Other arguments explicitly specified on the command prompt overwrite the corresponding configuration file settings.

-D scripts\_directory

Overwrites the default saved script directory specified in the registry or the configuration file specified with **-C**. The value does not persist in the registry or the configuration file. To see the current value of this option in SQL Query Analyzer, click **Tools**, and then click **Options**.

-**T** template\_directory

Overwrites the default template directory specified in the registry or the configuration file specified with **-C**. The value does not persist in the registry or the configuration file. To see the current value of this option in SQL Query Analyzer, click **Tools**, and then click **Options**.

# Remarks

The **isqlw** utility can be used with or without a user interface. To run **isqlw** without a user interface, specify valid login information (an instance of SQL Server 2000 with a trusted connection or a valid login ID and password) and input and output files. **isqlw** executes the contents of the input file and saves the results in the output file.

If input and output files are not specified, **isqlw** runs interactively, starting the SQL Query Analyzer. If valid login information is specified, **isqlw** connects directly an instance of SQL Server 2000. If not enough information is specified to connect, the **Connect to SQL Server** dialog box appears.

**isqlw** and SQL Query Analyzer use the ODBC API. The utility uses the Microsoft® SQL Server ODBC driver default settings for SQL-92. For more information, see Effects of SQL-92 Options.

For more information about this utility, see <u>Getting Started with Command</u> <u>Prompt Utilities</u>.

## Examples

## A. Execute SQL statements

This example connects to **MyServer** (**pubs** database) and executes SQL statements from *input\_file* and stores the results of the execution in *output\_file*.

isqlw -S MyServer -d pubs -U sa -P -i input\_file -o output\_file

# **B.** Use wildcards

This example loads all .sql files into SQL Query Analyzer. All connections use Windows Authentication and point to the **pubs** database on the local server.

isqlw -d pubs -E -f "c:\Program Files\Microsoft SQL Server\MSSQL\I

# **C. Load multiple files**

This example loads two files into SQL Query Analyzer. It uses Windows Authentication to connect to the local server.

isqlw -d pubs -E -f "c:\Program Files\Microsoft SQL Server\MSSQL\I

## **D. Use Unicode files**

This example connects to the **pubs** database on **MyServer** using Windows Authentication and executes the Input.sql file. The results are saved in the Output.txt file. The files are opened as Unicode files.

isqlw -S MyServer -d pubs -E -i input.sql -o output.txt -FU

Command Prompt Utilities

# itwiz Utility

The **itwiz** utility allows the Index Tuning Wizard to be executed using a command prompt utility. The Index Tuning Wizard can also be started from SQL Server Enterprise Manager, SQL Query Analyzer, and SQL Profiler.

## Syntax

```
itwiz [-?]
  [
    -D database_name {-i workload_file | -t workload_trace_table_name}
    -o script_file_name
    [-S server_name[\instance]]
    {
       {-U login_id [-P password]}
       |-E
    }
    [-f tuning_feature_set]
    [-K keep_existing_indexes]
    [-M recommendation_quality]
    [-B storage_bound]
    [-n number_of_queries]
    [-C max_columns_in_index]
    [-T table_list_file]
    [-m minimum_improvement]
    [-F][-v]
  ]
```

# Arguments

-?

Displays usage information.

-D database\_name

Specifies the name of the database to be tuned.

### -i workload\_file

Specifies the name of the workload file to use as input for tuning. The file must be in one of these formats: .trc (SQL Profiler trace file), .sql (SQL file), or .log (SQL Server 7.0 trace file).

### -t workload\_table\_name

Specifies the name of a table containing the workload trace for tuning. The name is specified as: [*server\_name*].[*database\_name*].

[*owner\_name*].*table\_name*. The first three parameters are optional and can be omitted by marking their positions with a period. The table shows the default values for each.

Parameter	Default value
server_name	<i>server_name</i> specified with –S option. If the – S option is not specified, <i>server_name</i> defaults to the local computer.
database_name	<i>database_name</i> specified with –D option.
owner_name	dbo.
table_name	None.

**Note** *owner\_name* must be dbo. If any other value is specified, execution of **itwiz** will fail and an error will be returned.

#### -o script\_file\_name

Specifies the name of the file to which **itwiz** writes the recommendation script. By default, output files are created in the current directory. The recommendation script contains the expected improvement if the recommendation is accepted.

### -S server\_name[\instance]

Specifies the computer and instance of SQL Server to connect to. If no *server\_name* or *instance* is specified, **itwiz** connects to the default instance of SQL Server on the local computer. This option is required when executing **itwiz** from a remote computer on the network.

-U login\_id

Specifies the login ID used to connect to SQL Server.

## -P password

Specifies the password for the login ID. If this option is not used, **itwiz** prompts for a password. If this option is used without specifying a password, **itwiz** uses the default password (NULL).

### -E

Uses a trusted connection instead of requesting a password.

### -f tuning\_feature\_set

Specifies the features to be considered by **itwiz** for tuning.

Value	Description
0	All features (default)
1	Indexes only
2	Indexed views only (applies only to SQL
	Server 2000, Enterprise and Developer
	editions)

### -K keep\_existing\_indexes

Specifies whether **itwiz** is allowed to propose a recommendation that requires dropping one or more existing indexes.

Value	Description
0	Do not keep existing indexes
1	Keep all existing indexes (default)

### -M recommendation\_quality

Specifies the desired point in the running time versus quality of recommendation tradeoff. Higher values of *recommendation\_quality* yield better quality of recommendation. Currently, *recommendation\_quality* can be

one of the values shown in this table.

Value	Description
0	Fast mode
1	Medium mode (default)
2	Thorough analysis mode

Fast mode currently has these restrictions:

- No new clustered indexes are recommended.
- No new indexed views are recommended.
- All existing indexes are kept (this is equivalent to specifying the **-K 1** option).

**Note** The combinations **-M 0 -K 0** and **-M 0 -f 2** are invalid and cannot be used. Also, when used in conjunction with **-M 0**, options **-f 0** and **-f 1** are equivalent.

### -B storage\_bound

Specifies the maximum space in megabytes that can be consumed by the recommended index set. The default storage bound is three times the current data size or the maximum available space on all attached disk drives, or whichever is smaller. The current data size consists of all tables and clustered indexes.

### -n number\_of\_queries

Specifies the number of queries to be tuned. By default, 200 queries are randomly chosen from the specified workload file. If *number\_of\_queries* exceeds the number of queries in the workload file, all queries are tuned.

### -C max\_columns\_in\_index

Specifies the maximum number of columns in indexes proposed by **itwiz**. The default value is 16; this is the maximum value allowed by SQL Server.

### -T table\_list\_file

Specifies the name of a file containing a list of tables to be tuned. Each table listed within the file should begin on a new line. Table names can be qualified by a user name (for example, **dbo.authors**). Optionally, to invoke the table-scaling feature, the name of a table can be followed by a number indicating the projected number of rows in the table. The table-scaling feature enables studying recommended indexes on smaller scale sample databases. A reasonable size (several %, thousands of rows per table) should be used for the smaller sample database, otherwise the scaled data distribution histograms may be inaccurate and the set of recommended indexes for the sample database may be different from the index recommended for the full scale database.

This is the file format for *table\_list\_file*:

[owner.]table [number\_of\_rows] [owner.]table [number\_of\_rows]

••• •••

If the **-T** option is omitted, all user tables in the specified database are considered for tuning.

### -m minimum\_improvement

If the **-m** option is specified, **itwiz** does not recommend any changes in the index configuration, unless the expected improvement in performance for the selected workload is at least *minimum\_improvement%*. If all queries are not considered for tuning (see option **-n**), the queries not selected are not considered when the improvement is evaluated.

#### -F

Permits **itwiz** to overwrite an existing output file. In the event that an output file with the same name already exists and **-F** is not specified, **itwiz** returns an error.

#### -v

Enables verbose output from **itwiz**. If **-v** is not specified, **itwiz** directs only abbreviated information to the screen during execution.

## Remarks

For more information about this utility, see <u>Getting Started with Command</u> <u>Prompt Utilities</u>.

# Examples

## A. Tune remote instances

This example connects to the **tpcd1G** database on remote server **autoadmin5**. The connection uses a login ID of **sa**, and a password of NULL.

itwiz –S autoadmin5 -U sa –P -D tpcd1G -i tpcd\_22.sql –o script.sql

## **B.** Limit disk use

This example limits the size of newly created indexes to 3 gigabytes and directs the output to **d:\result\_dir\script1.sql**.

itwiz –D tpcd1G –i tpcd\_22.sql -B 3000 –o "d:\result\_dir\script1.sql"

## C. Limit the number of tuned queries

This example limits the number of queries read from file **orders\_wkld.sql** to a maximum of 10.

```
itwiz –D orders –i orders_wkld.sql –o script.sql -n 10
```

## **D.** Tune specific tables

This example demonstrates the use of a *table\_list\_file* (**-T** option). The contents of **table\_list.txt** specifies that:

• Only the **authors**, **titles**, and **publishers** tables in the database should be tuned.

- The number of rows in the **authors** and **titles** tables should be assumed to be 100,000 and 2,000,000 respectively.
- The number of rows in **publishers** should be assumed to be the current number of rows in the table.

itwiz –D pubs –i pubs\_wkld.sql –o script.sql –T table\_list.txt

Here is the content of file **table\_list.txt**:

dbo.authors 100000 dbo.publishers titles 2000000

## E. Use thorough analysis mode

This example shows using the thorough analysis mode (-**M** option).

itwiz –D tpcd1G –i tpcd\_22query.sql –M 2 –E –o tpcd\_22recomm.sql

## See Also

**Index Tuning Wizard** 

Command Prompt Utilities

# makepipe Utility

The **makepipe** utility tests the integrity of the network Named Pipe services, in conjunction with **readpipe**.

## Syntax

makepipe [/h] [/w] [/p pipe\_name]

# Arguments

## /h

Displays usage information.

/w

Specifies the wait time, in seconds, between read and write. The default is 0.

/**p** pipe\_name

Is the name of the pipe. The default *pipe\_name* is abc.

# Remarks

This diagnostic utility checks to see if the named pipes are working. Use it (along with **readpipe**) if you cannot connect to an instance of Microsoft® SQL Server<sup>™</sup> 2000.

The **makepipe** utility is not installed by SQL Server 2000 Setup. It can be found in the *x*:\x86\Binn directory on the SQL Server 2000 compact disc.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## See Also

readpipe Utility

Command Prompt Utilities

# odbccmpt Utility

The **odbccmpt** utility enables or disables the compatibility option for an ODBC application executable file.

## Syntax

odbccmpt file\_name [/v:version\_number] [/P] [/T] [/R] [/d]

# Arguments

### file\_name

Is the name of the application executable file for which the compatibility option is to be turned on/off. For example, if the full path and name of your executable file is C:\Winnt\System32\MyApp.exe, only specify **MyApp**. Do not enclose *file\_name* in quotation marks.

Do not specify the name of the executable file for:

- Any application in the Microsoft SQL Server 2000 program group.
- Any SQL Server 2000 command prompt utility.
- The SQL Server executable file, Sqlservr.exe.

### /v:version\_number

Specifies that the application should be run in 6.x or 7.0 compatibility mode. Allowable values for *version\_number* are **6** and **7**. Invoking this option requests the server to communicate with the application using a lower level protocol.

### **/P**

Requires a connection peek call during SQL\_ATTR\_CONNECTION\_DEAD. Excluding this option provides a small performance advantage for applications that use connection pooling.

## **/T**

Specifies that translations be turned off in the specified application. This option provides backward compatibility with applications that use **SQLDriverConnect** and have no way to customize their connection properties.

### /**R**

Specifies that the application should ignore calls for a connection reset. When a connection returns to the connection pool, the driver is called with SQL\_COPT\_SS\_RESET. If this option is set, the application will ignore the call to reset the connection.

### /**d**

Disables the specified option for specified application. For example, odbccmpt /P /d will disable connection peek.

## Remarks

The 6.5/7.0 ODBC compatibility option sets certain behaviors of the 3.7 SQL Server ODBC driver to be compatible with earlier SQL Server ODBC drivers. This is only needed when using the 3.7 SQL Server ODBC driver and connecting to an instance of SQL Server 2000 using applications developed with earlier versions of the SQL Server ODBC driver. The 6.5/7.0 ODBC compatibility option can be turned on for applications using either the ODBC 2.*x* or ODBC 3.*x* API.

**odbccmpt** does not need to be run for applications connecting to earlier versions of SQL Server using the 3.7 SQL Server ODBC driver. The 3.7 driver automatically runs in compatibility mode when it connects to earlier versions of SQL Server. **odbccmpt** does not need to be run for applications using earlier versions of the SQL Server ODBC driver connecting to SQL Server 2000. SQL Server 2000 automatically treats any application using the earlier drivers as a 6.5 or 7.0-level application.

Enabling the 6.5 ODBC compatibility option for an application causes the application to be treated as a 6.5-level application. The setting enables these changes in the behavior of the 3.7 SQL Server ODBC driver when it connects to an instance of SQL Server 2000:

- The driver exhibits SQL Server 6.5 behavior for **SQLPrepare** and **SQLExecute**. The driver does not use the prepare/execute support built into SQL Server 2000. The driver generates temporary stored procedures if the option to generate stored procedures for **SQLPrepare** is turned on.
- The SET CONCAT\_NULL\_YIELDS\_NULL option is turned off.
- An input parameter can be bound as either SQL\_PARAM\_INPUT\_OUTPUT or SQL\_PARAM\_OUTPUT with **SQLBindParameter**.
- Result sets are returned by the driver in a format comparable to the result sets returned to a SQL Server 6.5 client:
  - Data from **nchar** and **nvarchar** columns are not treated as Unicode data. Instead, these columns are interpreted as character data using the Windows code page on the server running SQL Server. The ODBC catalog functions report **nchar**, **nvarchar**, and **uniqueidentifier** columns as **char**, **varchar**, and **varbinary** respectively.
  - Data cannot be retrieved from **ntext** columns.
  - **nchar** and **nvarchar** columns bound to SQL\_C\_DEFAULT return character data instead of Unicode data.
  - **char**, **varchar**, **binary**, and **varbinary** data values over 255 bytes are truncated to 255 bytes when they are returned to the application. **nchar** and **nvarchar** data values over 255 characters are truncated to 255 **char** or **varchar** characters.
  - Object names longer than 30 bytes are truncated to 30 bytes.

This truncation may affect additional operations, such as preventing **SQLSetPos** updates referencing **text** and **image** columns.

• Null values in bit columns are returned as 0.

Catalog results show all bit columns as NOT NULL because bit columns did not support null values in SQL Server 6.5.

- Data from **uniqueidentifier** columns is returned as **varbinary** data.
- The final node in the version string returned by the SQL\_DBMS\_VER parameter of **SQLGetInfo** never exceeds 255, regardless of the actual version string returned from the instance of SQL Server the application is connected to.
- The **SQLGetInfo** attributes reporting the data types supported by the driver do not report the new data types introduced after SQL Server 6.5.

If 7.0 compatibility is set, the server will communicate with the ODBC driver using 7.0 protocol. This means that the performance enhancements included in the prepare/execute model will not used. All new data types introduced with SQL Server 2000 (for example, **sql\_variant** and **bigint**) will be converted to the closest matching type that the 7.0 driver supports.

The ODBC compatibility option can only be set at the level of an executable file. It cannot be set for individual DLLs.

When **odbccmpt** is run to enable the 6.5 ODBC compatibility option, it creates a registry key:

```
HKEY_LOCAL_MACHINE
SOFTWARE
Microsoft
MSSQLServer
Client
```

ODBCAppCompat file\_name:REG\_SZ:6.5

When **odbccmpt** is run with the **-d** parameter, this key is removed. No error message is returned if **odbccmpt** is run with the **-d** parameter and there is no registry key for *file\_name*.

Whenever the 3.7 SQL Server ODBC driver connects to SQL Server 2000, it checks the registry for a key matching the name of the executable file for the application. If a match is found, the driver enables the 6.5 ODBC compatibility option. If no match is made, the 6.5 ODBC compatibility option is not enabled.

**Note** Most ODBC applications can switch without change from the 2.65 SQL Server ODBC driver to the 3.7 driver. **odbccmpt** is a migration aid to assist with moving applications from SQL Server 6.5 to SQL Server 2000. It should be used only for applications having problems in moving to SQL Server 2000 and the 3.7 SQL Server ODBC driver. **odbccmpt** should only be used until the application has been corrected.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# Examples

# A. Enable the ODBC compatibility option

odbccmpt MyApp /T odbccmpt MyApp /v:6

# **B.** Disable the ODBC compatibility option

odbccmpt MyApp /T /d

Command Prompt Utilities

# odbcping Utility

The **odbcping** utility tests the integrity of an ODBC data source and the ability of the client to connect to a server.

# Syntax

# Arguments

/?

Displays the **odbcping** syntax diagram.

-Sserver\_name

Is the instance of Microsoft® SQL Server<sup>™</sup> 2000 to connect to. Specify *server\_name* to connect to the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. The connection is made without testing any ODBC data source.

-Ddata\_source

Is the name of an ODBC data source defined to use the SQL Server ODBC driver. **odbcping** verifies that the data source is correct by using it to connect to the server named in the data source.

## -Ulogin\_id

Is a valid login ID for the server.

-Ppassword

Is the password for *login\_id*.

# Remarks

If the connection is successful, **odbcping** displays the version of the SQL Server ODBC driver and the version of the instance of SQL Server to which it connected.

If the connection attempt fails, **odbcping** displays the error messages it receives from the SQL Server ODBC driver.

The **odbcping** utility is not installed by SQL Server 2000 Setup. The utility can be found in the *x*:\x86\Binn directory on the SQL Server 2000 compact disc.

# Examples

# A. Verify connecting through a data source by using odbcping

odbcping /DMyDSN /Usa /PsaPassWord

# B. Verify connecting directly to a server by using odbcping

odbcping /SMyServer /Usa /PsaPassWord

See Also

Connecting to a SQL Server Data Source

Command Prompt Utilities

# osql Utility

The **osql** utility allows you to enter Transact-SQL statements, system procedures, and script files. This utility uses ODBC to communicate with the server.

# Syntax

```
osql
  [-?]|
  [-L]|
  [
     {
       {-U login_id [-P password]}
       |-E
     }
    [-S server_name[\instance_name]] [-H wksta_name] [-d db_name]
    [-l time_out] [-t time_out] [-h headers]
    [-s col_separator] [-w column_width] [-a packet_size]
    [-e] [-I] [-D data_source_name]
    [-c cmd_end] [-q "query"] [-Q "query"]
    [-n] [-m error_level] [-r {0 | 1}]
    [-i input_file] [-o output_file] [-p]
    [-b] [-u] [-R] [-O]
  ]
```

## Arguments

## -?

Displays the syntax summary of **osql** switches.

### -L

Lists the locally configured servers and the names of the servers broadcasting on the network.

-U login\_id

Is the user login ID. Login IDs are case-sensitive.

-P password

Is a user-specified password. If the **-P** option is not used, **osql** prompts for a password. If the **-P** option is used at the end of the command prompt without any password, **osql** uses the default password (NULL). Passwords are case-sensitive.

The OSQLPASSWORD environment variable allows you to set a default password for the current session. Therefore, you do not have to hard code a password into batch files.

If you do not specify a password with the **-P** option, **osql** first checks for the OSQLPASSWORD variable. If no value is set, **osql** uses the default password, NULL. The following example sets the OSQLPASSWORD variable at a command prompt and then accesses the **osql** utility:

## C:\>SET OSQLPASSWORD=abracadabra C:\>osql

## -E

Uses a trusted connection instead of requesting a password.

-S server\_name[\instance\_name]

Specifies the instance of Microsoft® SQL Server<sup>™</sup> 2000 to connect to. Specify *server\_name* to connect to the default instance of SQL Server on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, **osql** connects to the default instance of SQL Server on the local computer. This option is required when executing **osql** from a remote computer on the network.

### -H wksta\_name

Is a workstation name. The workstation name is stored in **sysprocesses.hostname** and is displayed by **sp\_who**. If this option is not specified, the current computer name is assumed.

### **-d** *db\_name*

Issues a USE *db\_name* statement when **osql** is started.

### -l time\_out

Specifies the number of seconds before an **osql** login times out. The default time-out for login to **osql** is eight seconds.

### -t time\_out

Specifies the number of seconds before a command times out. If a *time\_out* value is not specified, commands do not time out.

### -h headers

Specifies the number of rows to print between column headings. The default is to print headings one time for each set of query results. Use -1 to specify that no headers will be printed. If –1 is used, there must be no space between the parameter and the setting (**-h-1**, not **-h -1**).

### -s col\_separator

Specifies the column-separator character, which is a blank space by default. To use characters that have special meaning to the operating system (for example, |; & < >), enclose the character in double quotation marks (").

#### -w column\_width

Allows the user to set the screen width for output. The default is 80 characters. When an output line has reached its maximum screen width, it is broken into multiple lines.

### -a packet\_size

Allows you to request a different-sized packet. The valid values for *packet\_size* are 512 through 65535. The default value **osql** is the server default. Increased packet size can enhance performance on larger script execution where the amount of SQL statements between GO commands is substantial. Microsoft testing indicates that 8192 is typically the fastest setting for bulk copy operations. A larger packet size can be requested, but **osql** defaults to the server default if the request cannot be granted.

-е

Echoes input.

-I

Sets the QUOTED\_IDENTIFIER connection option on.

-D data\_source\_name

Connects to an ODBC data source that is defined using the ODBC driver for Microsoft SQL Server. The **osql** connection uses the options specified in the data source.

Note This option does not work with data sources defined for other drivers.

### -c cmd\_end

Specifies the command terminator. By default, commands are terminated and sent to SQL Server 2000 by entering GO on a line by itself. When you reset the command terminator, do not use Transact-SQL reserved words or characters that have special meaning to the operating system, whether preceded by a backslash or not.

## -q "query"

Executes a query when **osql** starts, but does not exit **osql** when the query completes. (Note that the query statement should not include GO). If you issue a query from a batch file, use %variables, or environment %variables%. For example:

```
SET table = sysobjects
osql /q "Select * from %table%"
```

Use double quotation marks around the query and single quotation marks around anything embedded in the query.

## -Q "query"

Executes a query and immediately exits **osql**. Use double quotation marks around the query and single quotation marks around anything embedded in the query.

-n

Removes numbering and the prompt symbol (>) from input lines.

-m error\_level

Customizes the display of error messages. The message number, state, and error level are displayed for errors of the specified severity level or higher. Nothing is displayed for errors of levels lower than the specified level. Use **-1** to specify that all headers are returned with messages, even informational messages. If using **-1**, there must be no space between the parameter and the setting (**-m-1**, not **-m -1**).

### -r $\{0 \mid 1\}$

Redirects message output to the screen (**stderr**). If you do not specify a parameter, or if you specify **0**, only error messages with a severity level 17 or higher are redirected. If you specify **1**, all message output (including "print") is redirected.

### -i input\_file

Identifies the file that contains a batch of SQL statements or stored procedures. The less than (<) comparison operator can be used in place of **-i**.

### -o output\_file

Identifies the file that receives output from **osql**. The greater than (>) comparison operator can be used in place of **-o**.

If *input\_file* is not Unicode and **-u** is not specified, *output\_file* is stored in OEM format. If *input\_file* is Unicode or **-u** is specified, *output\_file* is stored in Unicode format.

### -p

Prints performance statistics.

### -b

Specifies that **osql** exits and returns a DOS ERRORLEVEL value when an error occurs. The value returned to the DOS ERRORLEVEL variable is 1 when the SQL Server error message has a severity of 10 or greater; otherwise, the value returned is 0. Microsoft MS-DOS® batch files can test the value of DOS ERRORLEVEL and handle the error appropriately.

#### -u

Specifies that *output\_file* is stored in Unicode format, regardless of the format of the *input\_file*.
#### -R

Specifies that the SQL Server ODBC driver use client settings when converting currency, date, and time data to character data.

#### -0

Specifies that certain **osql** features be deactivated to match the behavior of earlier versions of **isql**. These features are deactivated:

- EOF batch processing
- Automatic console width scaling
- Wide messages

It also sets the default DOS ERRORLEVEL value to -1.

### Remarks

The **osql** utility is started directly from the operating system with the casesensitive options listed here. After **osql** starts, it accepts SQL statements and sends them to SQL Server interactively. The results are formatted and displayed on the screen (**stdout**). Use QUIT or EXIT to exit from **osql**.

If you do not specify a user name when you start **osql**, SQL Server 2000 checks for the environment variables and uses those, for example, **osqluser=**(*user*) or **osqlserver=**(*server*). If no environment variables are set, the workstation user name is used. If you do not specify a server, the name of the workstation is used.

If neither the **-U** or **-P** options are used, SQL Server 2000 attempts to connect using Windows Authentication Mode. Authentication is based on the Microsoft Windows NT® account of the user running **osql**.

The **osql** utility uses the ODBC API. The utility uses the SQL Server ODBC driver default settings for the SQL Server 2000 SQL-92 connection options. For more information, see <u>Effects of SQL-92 Options</u>.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# **OSQL** Commands

In addition to Transact-SQL statements within **osql**, these commands are also available.

Command	Description
GO	Executes all statements entered after the last GO.
RESET	Clears any statements you have entered.
ED	Calls the editor.
<b>!!</b> command	Executes an operating-system command.
QUIT or EXIT()	Exits from <b>osql</b> .
CTRL+C	Ends a query without exiting from <b>osql</b> .

The command terminators GO (by default), RESET, ED, **!!**, EXIT, QUIT, and CTRL+C, are recognized only if they appear at the beginning of a line, immediately following the **osql** prompt. Anything entered on the same line after these keywords is disregarded by **osql**.

GO signals both the end of a batch and the execution of any cached Transact-SQL statements. When you press ENTER at the end of each input line, **osql** caches the statements on that line. When you press ENTER after typing GO, all of the currently cached statements are sent as a batch to SQL Server.

The current **osql** utility works as if there is an implied GO at the end of any script executed, therefore all statements in the script execute. Any statements after the last GO are not executed.

End a command by typing a line beginning with a command terminator. You can follow the command terminator with an integer to specify how many times the command should be run. For example, to execute this command 100 times, type:

SELECT x = 1 GO 100

The results are printed once at the end of execution. **osql** does not accept more than 1,000 characters per line. Large statements should be spread across multiple lines.

The user can call an editor on the current query buffer by typing ED as the first word on a line. The editor is defined in the EDITOR environment variable. The default editor is "edit." You can specify a different editor by setting the EDITOR environment variable. For example, to make the default editor Notepad, enter at the operating-system prompt:

SET EDITOR=notepad

# **Operating-System Commands**

Operating-system commands can also be executed by starting a line with two exclamation points (!!) followed by the command. The command recall facilities of Windows NT can be used to recall and modify **osql** statements. The existing query buffer can be cleared by typing RESET.

When running stored procedures, **osql** prints a blank line between each set of results in a batch. In addition, the "0 rows affected" message does not appear when it does not apply to the statement executed.

# Using osql Interactively

To use **osql** interactively, type the **osql** command (and any of the options) at a command prompt.

You can read in a file containing a query (such as Stores.qry) for execution by **osql** by typing a command similar to this:

osql /U alma /P /i stores.qry

The file must include a command terminator(s).

You can read in a file containing a query (such as Titles.qry) and direct the results to another file by typing a command similar to this:

osql /U alma /P /i titles.qry /o titles.res

When using **osql** interactively, you can read an operating-system file into the command buffer with **:r** *file\_name*. Do not include a command terminator in the file; enter the terminator interactively after you have finished editing.

# **Inserting Comments**

You can include comments in a Transact-SQL statement submitted to SQL Server by **osql**. Two types of commenting styles are allowed: -- and /\*...\*/.

For more information, see <u>Using Comments</u>.

# Using EXIT to Return Results in osql

You can use the result of a SELECT statement as the return value from **osql**. The first column of the first result row is converted to a 4-byte integer (long). MS-DOS passes the low byte to the parent process or operating system error level. Windows NT passes the entire 4-byte integer. The syntax is:

EXIT(query)

For example:

```
EXIT(SELECT @@ROWCOUNT)
```

EXIT(SELECT 5)

You can also include the EXIT parameter as part of a batch file. For example:

# osql /Q "EXIT(SELECT COUNT(\*) FROM '%1')"

The **osql** utility passes everything between the parentheses () to the server exactly as entered. If a stored system procedure selects a set and returns a value, only the selection is returned. The EXIT() statement with nothing between the parentheses executes everything preceding it in the batch and then exits with no return value.

There are four EXIT formats:

• EXIT

Does not execute the batch; quits immediately and returns no value.

• EXIT()

Executes the batch, and then quits and returns no value.

• EXIT(query)

Executes the batch, including the query, and then quits after returning the results of the query.

• RAISERROR with a state of 127

If RAISERROR is used within an **osql** script and a state of 127 is raised, **osql** will quit and return the message ID back to the client. For example:

RAISERROR(50001, 10, 127)

This error will cause the **osql** script to end and the message ID 50001 will be returned to the client.

The return values -1 to -99 are reserved by SQL Server; **osql** defines these values:

• -100

Error encountered prior to selecting return value.

• -101

No rows found when selecting return value.

• -102

Conversion error occurred when selecting return value.

# **Displaying money and smallmoney Data Types**

**osql** displays the **money** and **smallmoney** data types with two decimal places although SQL Server stores the value internally with four decimal places. Consider the example:

SELECT CAST(CAST(10.3496 AS money) AS decimal(6, 4))

This statement produces a result of 10.3496, which indicates that the value is stored with all decimal places intact.

# See Also

/\*...\*/ (Comment) -- (Comment) CAST and CONVERT Managing Security RAISERROR Command Prompt Utilities

# **Rebuild master Utility**

The Rebuild **master** (**rebuildm**) utility can be used to change the collation settings for an instance of Microsoft® SQL Server<sup>™</sup> 2000, or to fix a corrupted **master** database.

# **Syntax**

rebuildm

# Remarks

Before you run the Rebuild **master** utility, make sure you have one of these two items:

- The scripts to rebuild the database objects and a backup to reload the data.
- The data and log files to use with **sp\_attach\_db**.

Rebuilding the **master** database removes all database objects and data. After rebuilding the **master** database, re-create the database objects and reload the data, or reattach the data and log files using **sp\_attach\_db**.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# See Also

How to rebuild the master database (Rebuild Master utility)

Command Prompt Utilities

# readpipe Utility

The **readpipe** utility tests the integrity of the network Named Pipe services, in conjunction with **makepipe**.

# **Syntax**

readpipe /Sserver\_name /Dstring [/n] [/q] [/w] [/t] [/p pipe\_name] [/h]

# Arguments

/Sserver\_name

Is the name of a default instance of Microsoft® SQL Server<sup>™</sup> on which you just ran **makepipe**. **readpipe** does not support named instances of SQL Server 2000. There are no spaces between /**S** and *server\_name*.

#### /**D**string

Is a test character string. There are no spaces between /**D** and *string*.

#### /n

Specifies the number of iterations.

#### /q

Queries for incoming data (polling). Without /**q**, **readpipe** reads the pipe and waits for data.

#### /w

Specifies the wait time, in seconds, to pause while polling. The default is 0.

#### /t

Asks for Transact-SQL named pipes. This option overrides polling.

#### /**p** pipe\_name

Is the name of the pipe. The default *pipe\_name* is abc.

#### /h

Displays usage.

# Remarks

This diagnostic utility checks to see if the named pipes are working. Use it (along with **makepipe**) if you cannot connect to an instance of SQL Server.

Because applications for Microsoft Windows NT®, Microsoft Windows®, and Microsoft MS-DOS® all run on Windows NT, you must be careful about which version of **makepipe** and **readpipe** you are running. You must use the version specific to the operating system on which you are testing the integrity of the named pipe.

After **makepipe** is started, the server waits for a client to connect. The **readpipe** command prompt utility can then be run from other workstations. When all testing is complete, go to the screen where the **makepipe** utility is running and press CTRL+BREAK or CTRL+C to stop execution.

The **readpipe** utility is not installed by SQL Server Setup. The utility can be found in the *x*:\x86\Binn directory on the SQL Server compact disc.

See Also

makepipe Utility

Command Prompt Utilities

# **Replication Distribution Agent Utility**

The Replication Distribution Agent utility configures and begins the Distribution Agent, which moves the snapshot (for snapshot replication and transactional replication) held in the distribution database tables (for transactional replication) to the destination tables at the Subscribers.

# Syntax

distrib [-?] -**Publisher** server name[\instance name] -PublisherDB publisher\_database -Subscriber server\_name[\instance\_name] [-BcpBatchSize bcp batch size] [**-Buffers** number\_of\_buffers] [-CommitBatchSize commit batch size] [-CommitBatchThreshold commit\_batch\_threshold] [-Continuous] [**-DefinitionFile** *def\_path\_and\_file\_name*] [**-Distributor** *distributor*] [-DistributorAddress distributor address] [-DistributorLogin distributor\_login] [-DistributorNetwork distributor\_network] [**-DistributorPassword** *distributor\_password*] [-DistributorSecurityMode [0|1]] [**-ErrorFile** *error\_path\_and\_file\_name*] [-FileTransferType [0|1]] [-**FtpAddress** *ftp address*] [-**FtpPassword** *ftp password*] [-**FtpPort** *ftp\_port*] [-FtpUserName ftp\_user\_name] [-HistoryVerboseLevel [1|2|3]] [**-LoginTimeOut** login\_time\_out\_seconds] [-MaxBcpThreads] [-MaxDeliveredTransactions number of transactions] [-MessageInterval message\_interval]

```
[-NoTextInitOnSync]
[-Output output_path_and_file_name]
[-OutputVerboseLevel [0|1|2]]
[-PacketSize packet size]
[-PollingInterval polling interval]
[-ProfileName profile_name]
[-Publication publication]
[-QueryTimeOut query_time_out_seconds]
[-QuotedIdentifier quoted_identifier]
[-SubscriberDatabasePath subscriber path]
[-SubscriberDB subscriber database]
[-SubscriberLogin subscriber login]
[-SubscriberPassword subscriber password]
[-SubscriberSecurityMode [0|1]]
[-SubscriberType [0|1|2|3]]
[-SubscriptionTableName subscription_table]
[-SubscriptionType [0|1|2]]
[-TransactionsPerHistory [0|1|...10000]]
```

**Note** Parameters can be specified in any order. When optional parameters are not specified, values from predefined registry settings on the local computer are used.

# Arguments

# -?

Prints all available parameters.

-Publisher server\_name[\instance\_name]:

Is the name of the Publisher. Specify *server\_name* for the default instance of Microsoft® SQL Server<sup>™</sup> 2000 on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

-PublisherDB publisher\_database:

Is the name of the Publisher database.

-Subscriber server\_name[\instance\_name]:

Is the name of the Subscriber. Specify *server\_name* for the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

#### -BcpBatchSize bcp\_batch\_size:

Is the number of rows to send in a bulk copy operation. When performing a **bcp in** operation, the batch size is the number of rows to send to the server as one transaction, and also the number of rows that must be sent before the Distribution Agent logs a **bcp** progress message. When performing a **bcp out** operation, a fixed batch size of **1000** is used. A value of **0** indicates no message logging.

### -Buffers number\_of\_buffers:

Is the number of buffers available for asynchronous transactions. The default is 2. Increasing the number of buffers can increase performance because doing so reduces memory paging; however, a larger number of buffers also increases the amount of memory reserved for paging. Performance can be evaluated by seeing how a change in buffer values affects the speed of the connections between Subscriber and Distributor.

#### -CommitBatchSize commit\_batch\_size:

Is the number of transactions to be issued to the Subscriber before a COMMIT statement is issued. The default is 100.

#### -CommitBatchThreshold commit\_batch\_threshold:

Is the number of replication commands to be issued to the Subscriber before a COMMIT statement is issued. The default is 1000.

#### -Continuous:

Specifies whether the agent attempts to poll replicated transactions continually. If specified, the agent polls replicated transactions from the source at polling intervals, even if there are no transactions pending.

#### -DefinitionFile def\_path\_and\_file\_name:

Is the path of the agent definition file. An agent definition file contains command prompt arguments for the agent. The content of the file is parsed as an executable file. Double quotation marks (") can be used to specify

argument values containing arbitrary characters.

#### -Distributor distributor:

Is the Distributor name. For Distributor (push) distribution, the name defaults to the name of the local Distributor.

### -DistributorAddress distributor\_address

Is the network connection string for the Net-Library defined in the **DistributorNetwork** option. If the **DistributorNetwork** option is the TCP/IP Sockets Net-Library, then the connection string is in the form of

#### 'address,socket'

For more information about the format, see <u>Network Protocols</u>. This option is useful for configuring connections over the Internet.

#### -DistributorLogin distributor\_login:

Is the Distributor login name.

#### -DistributorNetwork distributor\_network:

Is the Net-Library to use when connecting to the Distributor. This option is useful when configuring the Distribution Agent to connect to a Distributor over the Internet.

#### -DistributorPassword distributor\_password:

Is the Distributor password.

# -DistributorSecurityMode [0|1]:

Specifies the security mode of the Distributor. A value of 0 indicates SQL Server Authentication Mode (default), and a value of 1 indicates Windows Authentication Mode.

# -ErrorFile error\_path\_and\_file\_name:

Is the path and file name of the error file generated by the Distribution Agent. This file is generated at any point where failure occurred while applying replication transactions at the Subscriber. This file contains the failed replication transactions and associated error messages. When not specified, the error file is generated in the current directory of the Distribution Agent. The error file name is the name of the Distribution Agent with an .err extension. If the specified file name exists, error messages are appended to the file.

### -FileTransferType [0|1]

Specifies the file transfer type. A value of **0** indicates UNC (universal naming convention), and a value of **1** indicates FTP (file transfer protocol).

#### -FtpAddress ftp\_address

Is the network address of the FTP service for the Distributor. When not specified, **DistributorAddress** is used. If **DistributorAddress** is not specified, **Distributor** is used.

#### -FtpPassword ftp\_password

Is the user password used to connect to the FTP service.

#### -FtpPort ftp\_port

Is the port number of the FTP service for the Distributor. When not specified, the default port number for FTP service (21) is used.

#### -FtpUserName ftp\_user\_name

Is the user name used to connect to the FTP service. When not specified, anonymous is used.

#### -HistoryVerboseLevel [1|2|3]

Specifies the amount of history logged during a distribution operation. You can minimize the performance affect of history logging by selecting **1**.

HistoryVerboseLevel	
value	Description
1	Default. Always update a previous history message of the same status (startup, progress, success, and so forth). If no previous record with the same status exists, insert a new record.
2	Insert new history records unless the record is for such things as idle messages or long-running job messages, in which case update the previous

	records.
3	Always insert new records, unless it is for idle
	messages.

#### -LoginTimeOut login\_time\_out\_seconds

Is the number of seconds before the login times out. The default is 15 seconds.

#### -MaxBcpThreads

Specifies the number of bulk copy operations that can be performed in parallel. The maximum number of threads and ODBC connections that exist simultaneously is the lesser of **MaxBcpThreads** or the number of bulk copy requests that appear in the synchronization transaction in the distribution database. **MaxBcpThreads** must have a value greater than zero, and has no hard-coded upper limit. The default is 1.

### -MaxDeliveredTransactions number\_of\_transactions

Is the maximum number of push or pull transactions applied to Subscribers in one synchronization. A value of **0** indicates that the maximum is an infinite number of transactions. Other values can be used by Subscribers to shorten the duration of a synchronization being pulled from a Publisher.

#### -MessageInterval message\_interval

Is the time interval used for history logging. A history event is logged when one of these parameters is reached:

- The **TransactionsPerHistory** value is reached after the last history event is logged.
- The **MessageInterval** value is reached after the last history event is logged.

If there is no replicated transaction available at the source, the agent reports a no-transaction message to the Distributor. This option specifies how long the agent waits before reporting another notransaction message. Agents always report a no-transaction message when they detect that there are no transactions available at the source after previously processing replicated transactions. The default is 3600 (seconds).

### -NoTextInitOnSync

Specifies that applications modifying **text** or **image** columns in the published table initialize pointers with some value other than NULL in the UPDATE statement.

#### -Output output\_path\_and\_file\_name:

Is the path of the agent output file. If the file name is not provided, the output is sent to the console. If the specified file name exists, the output is appended to the file.

#### -OutputVerboseLevel [0|1|2]:

Specifies whether the output should be verbose. If the verbose level is **0**, only error messages are printed. If the verbose level is **1**, all the progress report messages are printed. If the verbose level is **2** (default), all error messages and progress report messages are printed, which is useful for debugging.

#### -PacketSize packet\_size:

Is the packet size, in bytes. The default is 4096 (bytes).

#### -PollingInterval polling\_interval:

Is how often, in seconds, the distribution database is queried for replicated transactions. The default is 3 seconds.

#### -ProfileName profile\_name:

Specifies an agent profile to use for agent parameters. If **ProfileName** is NULL, the agent profile is disabled. If **ProfileName** is not specified, the default profile for the agent type is used.

#### -Publication publication:

Is the name of the publication. This parameter is only valid if the publication is set to always have a snapshot available for new or reinitialized subscriptions.

#### -QueryTimeOut query\_time\_out\_seconds:

Is the number of seconds before the query times out. The default is 300 seconds.

#### -QuotedIdentifier quoted\_identifier:

Specifies the quoted identifier character to use. The first character of the value indicates the value the Distribution Agent uses. If **QuotedIdentifier** is used with no value, the Distribution Agent uses a space. If **QuotedIdentifier** is not used, the Distribution Agent uses whatever quoted identifier the Subscriber supports.

#### -SubscriberDatabasePath subscriber\_database\_path:

Is the path to the Jet database (.mdb file) if **SubscriberType** is **2** (allows a connection to a Jet database without an ODBC DSN).

#### -SubscriberDB subscriber\_database:

Is the name of the Subscriber database:

#### -SubscriberLogin subscriber\_login:

Is the Subscriber login name. If **SubscriberSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

#### -SubscriberPassword subscriber\_password:

Is the Subscriber password. If **SubscriberSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

#### -SubscriberSecurityMode [0|1]:

Specifies the security mode of the Subscriber. A value of **0** indicates SQL Server Authentication (default), and a value of **1** indicates Windows Authentication Mode.

#### -SubscriberType [0|1|2|3]

Specifies the type of Subscriber connection used by the Distribution Agent.

SubscriberType value	Description
0	Microsoft SQL Server

1	ODBC data source
2	Jet database (direct connection)
3	OLE DB data source

#### -SubscriptionTableName subscription\_table

Is the name of the subscription table generated or used at the given Subscriber. When not specified, the **MSreplication\_subscription** table is used. Use this option for database management systems (DBMS) that do not support long file names.

# -SubscriptionType [0|1|2]

Specifies the subscription type for distribution. A value of **0** indicates a push subscription, a value of **1** indicates a pull subscription, and a value of **2** indicates an anonymous subscription.

#### -TransactionsPerHistory [0|1|...10000]

Specifies the transaction interval for history logging. If the number of committed transactions after the last instance of history logging is greater than this option, a history message is logged. The default is 100. A value of **0** indicates infinite **TransactionsPerHistory**. See the preceding – **MessageInterval** parameter.

# Remarks

**IMPORTANT** If you have installed SQL Server Agent to run under a local system account rather than under a domain user account (the default), the service can only access the local computer. If the Distribution Agent that runs under SQL Server Agent is configured to use Windows Authentication Mode when it logs in to an instance of SQL Server, the Distribution Agent fails. The default setting is SQL Server Authentication.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# See Also

Distribution Agent Profile

Replication Agents

Command Prompt Utilities

# **Replication Log Reader Agent Utility**

The Replication Log Reader Agent utility configures and begins the Log Reader Agent, which monitors the transaction log of each database configured for replication, and copies the transactions marked for replication from the transaction log into the distribution database.

# Syntax

logread [-?] -**Publisher** server name[\instance name] -PublisherDB publisher database [-Buffers number\_of\_buffers] [-Continuous] [**-DefinitionFile** *def\_path\_and\_file\_name*] [**-Distributor** server\_name[\instance\_name]] [-DistributorLogin distributor\_login] [**-DistributorPassword** *distributor\_password*] [-DistributorSecurityMode [0|1]] [-HistoryVerboseLevel [1|2]] [**-LoginTimeOut** login\_time\_out\_seconds] [-MessageInterval message\_interval] [**-Output** *output\_path\_and\_file\_name*] [-OutputVerboseLevel [0|1|2]] [**-PacketSize** packet\_size] [**-PollingInterval** polling\_interval] [-PublisherSecurityMode [0|1]] [-PublisherLogin publisher login] [-PublisherPassword publisher\_password] [-QueryTimeOut query\_time\_out\_seconds] [-ReadBatchSize number of transactions] [-ReadBatchThreshold read batch threshold]

**Note** Parameters can be specified in any order. When optional parameters are not specified, values from predefined registry settings on the local computer are used.

# Arguments

#### -?

Displays usage information.

#### -Publisher server\_name[\instance\_name]

Is the name of the Publisher. Specify *server\_name* for the default instance of Microsoft® SQL Server<sup>™</sup> 2000 on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

#### -PublisherDB publisher\_database

Is the name of the Publisher database.

### -Buffers number\_of\_buffers

Is the number of buffers available for asynchronous transactions. The default is 2. Increasing the number of buffers can increase performance because doing so reduces memory paging; however, a larger number of buffers also increases the amount of memory reserved for paging. Performance can be evaluated by seeing how a change in buffer values affects the speed of the connections between Subscriber and Distributor.

#### -Continuous

Specifies whether the agent attempts to poll replicated transactions continually. If specified, the agent polls replicated transactions from the source at polling intervals even if there are no transactions pending.

#### -DefinitionFile def\_path\_and\_file\_name

Is the path of the agent definition file. An agent definition file contains command prompt arguments for the agent. The content of the file is parsed as an executable file. Double quotation marks (") can be used to specify argument values containing arbitrary characters.

#### -Distributor server\_name[\instance\_name]

Is the Distributor name. Specify *server\_name* for the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

#### -DistributorLogin distributor\_login

Is the Distributor login name.

#### -DistributorPassword distributor\_password

Is the Distributor password.

### -DistributorSecurityMode [0|1]

Specifies the security mode of the Distributor. A value of **0** indicates SQL Server Authentication Mode (default), and a value of **1** indicates Windows Authentication Mode.

#### -HistoryVerboseLevel [1|2]

Specifies the amount of history logged during a log reader operation. You can minimize the performance affect of history logging by selecting **1**.

HistoryVerboseLevel	
value	Description
1	Default. Always update a previous history message of the same status (startup, progress, success, and so forth). If no previous record with the same status exists, insert a new record.
2	Insert new history records unless the record is for such things as idle messages or long-running job messages, in which case update the previous records.

#### -LoginTimeOut login\_time\_out\_seconds

Is the number of seconds before the login times out. The default is 15 seconds.

#### -MessageInterval message\_interval

Is the time interval used for history logging. A history event is logged when one of these parameters is reached:

• The **TransactionsPerHistory** value is reached after the last history

event is logged.

• The **MessageInterval** value is reached after the last history event is logged.

If there is no replicated transaction available at the source, the agent reports a no-transaction message to the Distributor. This option specifies how long the agent waits before reporting another notransaction message. Agents always report a no-transaction message when they detect that there are no transactions available at the source after previously processing replicated transactions. The default is 3600 (seconds).

#### -Output output\_path\_and\_file\_name

Is the path of the agent output file. If the file name is not provided, the output is sent to the console. If the specified file name exists, the output is appended to the file.

### -OutputVerboseLevel [0|1|2]

Specifies whether the output should be verbose. If the verbose level is **0**, only error messages are printed. If the verbose level is **1**, all the progress report messages are printed. If the verbose level is **2** (default), all error messages and progress report messages are printed, which is useful for debugging.

#### -PacketSize packet\_size

Is the packet size, in bytes. The default is 4096 (bytes).

#### -PollingInterval polling\_interval

Is how often, in seconds, the log is queried for replicated transactions. The default is 10 seconds.

# -PublisherSecurityMode [0|1]

Specifies the security mode of the Publisher. A value of **0** indicates SQL Server Authentication (default), and a value of **1** indicates Windows Authentication Mode.

#### -PublisherLogin publisher\_login

Is the Publisher login name.

#### -PublisherPassword publisher\_password

Is the Publisher password.

#### -QueryTimeOut query\_time\_out\_seconds

Is the number of seconds before the query times out. The default is 300 seconds.

#### -ReadBatchSize number\_of\_transactions

Is the maximum number of transactions read out of the source. For the Log Reader Agent, the source is the transaction log of the publishing database. For the Distribution Agent, the source is the distribution database. The default is 500.

### -ReadBatchThreshold read\_batch\_threshold

Is the number of replication commands to be read from the transaction log before being issued to the Subscriber by the Distribution Agent. The default is 100.

# Remarks

**IMPORTANT** If you installed SQL Server Agent to run under a local system account rather than under a domain user account (the default), the service can access only the local computer. If the Log Reader Agent that runs under SQL Server Agent is configured to use Windows Authentication Mode when it logs in to Microsoft® SQL Server<sup>TM</sup>, the Log Reader Agent fails. The default setting is SQL Server Authentication.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# See Also

Log Reader Agent Profile

**Replication Agents** 

Command Prompt Utilities

# **Replication Merge Agent Utility**

The Replication Merge Agent utility configures and begins the Merge Agent, which applies the initial snapshot held in the database tables to the Subscribers. It also merges incremental data changes that occurred at the Publisher after the initial snapshot was created, and reconciles conflicts either according to the rules you configure or using a custom resolver you create.

# Syntax

replmerg [-?] -**Publisher** server name[\instance name] -PublisherDB publisher\_database -Publication publication -Subscriber server name[\instance name] -SubscriberDB subscriber database [-Continuous] [**-DefinitionFile** *def\_path\_and\_file\_name*] [**-Distributor** server\_name[\instance\_name]] [-**DistributorAddress** distributor\_address] [-DistributorLogin distributor\_login] [-DistributorNetwork distributor\_network] [-DistributorPassword distributor\_password] [-DistributorSecurityMode [0|1]] [-DownloadGenerationsPerBatch download\_generations\_per\_batch] [-**DownloadReadChangesPerBatch** download\_read\_changes\_per\_batch] [**-DownloadWriteChangesPerBatch** download\_write\_changes\_per\_batch] [-ExchangeType [1|2|3]] [-FastRowCount [0|1]] [-FileTransferType [0|1]] [-FtpAddress ftp\_address] [-FtpPassword ftp\_password] [-**FtpPort** *ftp\_port*] [-FtpUserName ftp\_user\_name] [-HistoryVerboseLevel [1|2|3]] [-Hostname host name]

[-LoginTimeOut login\_time\_out\_seconds] [-Output] [-OutputVerboseLevel [0|1|2]] [-PollingInterval polling interval] [-**ProfileName** profile\_name] [-PublisherAddress publisher address] [-PublisherLogin publisher\_login] [-PublisherNetwork publisher\_network] [-PublisherPassword publisher\_password] [-PublisherSecurityMode [0|1]] [**-QueryTimeOut** *query\_time\_out\_seconds*] [-SubscriberDatabasePath subscriber path] [-SubscriberDBAddOption [0|1|2|3]] [-SubscriberLogin subscriber\_login] [-SubscriberPassword subscriber\_password [-SubscriberSecurityMode [0|1]] [-SubscriberType [0|1|2|3|4|5|6|7|8]] [-SubscriptionType [0|1|2]] [-UploadGenerationsPerBatch upload generations per batch] [-UploadReadChangesPerBatch upload\_read\_changes\_per\_batch] [-UploadWriteChangesPerBatch upload\_write\_changes\_per\_batch] [-Validate [0|1|2|3]] [-ValidateInterval validate\_interval]

**Note** Parameters can be specified in any order. When optional parameters are not specified, values from predefined registry settings on the local computer are used.

# Arguments

-?

Prints all available parameters.

# -Publisher server\_name[\instance\_name]:

Is the name of the Publisher. Specify *server\_name* for the default instance of Microsoft® SQL Server<sup>™</sup> 2000 on that server. Specify *server\_name*\*instance\_name* for a named instance of SQL Server 2000 on

that server.

#### -PublisherDB publisher\_database

Is the name of the Publisher database.

### -Publication publication

Is the name of the publication. This parameter is only valid if the publication is set to always have a snapshot available for new or reinitialized subscriptions.

# -Subscriber server\_name[\instance\_name]

Is the name of the Subscriber. Specify *server\_name* for the default instance of SQL Server on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

### -SubscriberDB subscriber\_database

Is the name of the Subscriber database.

### -Continuous

Specifies whether the agent attempts to poll replicated transactions continually. If specified, the agent polls replicated transactions from the source at polling intervals, even if there are no transactions pending.

# -DefinitionFile def\_path\_and\_file\_name

Is the path of the agent definition file. An agent definition file contains command prompt arguments for the agent. The content of the file is parsed as an executable file. Double quotation marks (") can be used to specify argument values containing arbitrary characters.

# -Distributor server\_name[\instance\_name]

Is the Distributor name. Specify *server\_name* for the default instance of SQL Server on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server. For Distributor (push) distribution, the name defaults to the name of the default instance of SQL Server on the local computer.

# -DistributorAddress distributor\_address

Is the network connection string for the Net-Library defined in the **DistributorNetwork** option. If the **DistributorNetwork** option is the TCP/IP Sockets Net-Library, then the connection string is in the form of:

#### 'address,socket'

For more information about the format, see <u>Network Protocols</u>. This option is useful for configuring connections across the Internet.

### -DistributorLogin distributor\_login

Is the Distributor login name.

# -DistributorNetwork distributor\_network

Is the Net-Library (without the .dll extension) to use when connecting to the Distributor. This option is useful when configuring the Merge Agent to connect to a Distributor over the Internet.

### -DistributorPassword distributor\_password

Is the Distributor password.

# -DistributorSecurityMode [0|1]

Specifies the security mode of the Distributor. A value of **0** indicates SQL Server Authentication Mode (default), and a value of **1** indicates Windows Authentication Mode.

# -DownloadGenerationsPerBatch download\_generations\_per\_batch

Is the number of generations to be processed in a single batch while downloading changes from the Publisher to the Subscriber. A generation is defined as a logical group of changes per article. The default for a reliable communication link is 100. The default for an unreliable communication link is 10.

# -DownloadReadChangesPerBatch download\_read\_changes\_per\_batch

Is the number of changes to be read in a single batch while downloading changes from the Publisher to the Subscriber. The default is 100.

#### -DownloadWriteChangesPerBatch download\_write\_changes\_per\_batch

Is the number of changes to be applied in a single batch while downloading

changes from the Publisher to the Subscriber. The default is 100.

# -ExchangeType [1|2|3]

Specifies the type of exchange. A value of **1** indicates a push exchange. A value of **2** indicates a pull exchange. A value of **3** indicates a bidirectional exchange (default).

# -FastRowCount [0|1]

Specifies what type of rowcount calculation method should be used for rowcount validation. A value of **1** (default) indicates the fast method. A value of **0** indicates the full rowcount method.

# -FileTransferType [0|1]

Specifies the file transfer type. A value of **0** indicates UNC (universal naming convention), and a value of **1** indicates FTP (file transfer protocol).

# -FtpAddress ftp\_address

Is the network address of the FTP service for the Distributor. When not specified, **DistributorAddress** is used. If the **DistributorAddress** is not specified, **Distributor** is used.

# -FtpPassword ftp\_password

Is the user password used to connect to the FTP service.

# -FtpPort ftp\_port

Is the port number of the FTP service for the Distributor. When not specified, the default port number for FTP service (21) is used.

# -FtpUserName ftp\_user\_name

Is the user name used to connect to the FTP service. When not specified, anonymous is used.

# -HistoryVerboseLevel [1|2|3]

Specifies the amount of history logged during a merge operation. You can minimize the performance affect of history logging by selecting **1**.

#### HistoryVerboseLevel

value	Description
1	Always update a previous history message of the
	same status (startup, progress, success, and so
	forth). If no previous record with the same status
	exists, insert a new record.
2	Default. Insert new history records unless the
	record is for such things as idle messages or long-
	running job messages. In those instances, update
	the previous records.
3	Always insert new records, unless it is for idle
	messages.

#### -Hostname host\_name

Is the network name of the local computer. The default is the local computer name.

#### -LoginTimeOut login\_time\_out\_seconds

Is the number of seconds before the login times out. The default is 15 seconds.

#### -Output output\_path\_and\_file\_name

Is the path of the agent output file. If the file name is not provided, the output is sent to the console. If the specified file name exists, the output is appended to the file.

# -OutputVerboseLevel [0|1|2]

Specifies whether the output should be verbose. If the verbose level is **0**, only error messages are printed. If the verbose level is **1**, all of the progress report messages are printed. If the verbose level is **2** (default), all error messages and progress report messages are printed, which is useful for debugging.

#### -PollingInterval polling\_interval

Is how often, in seconds, the Publisher or Subscriber is queried for data changes. The default is 60 seconds.

#### -ProfileName profile\_name

Specifies an agent profile to use for agent parameters. If **ProfileName** is NULL, the agent profile is disabled. If **ProfileName** is not specified, the default profile for the agent type is used.

#### -PublisherAddress publisher\_address

Is the network connection string for the Net-Library defined in the **PublisherNetwork** option. If the **PublisherNetwork** option is the TCP/IP Sockets Net-Library, the connection string is in the form of:

#### 'address,socket'

For more information about the format, see <u>Network Protocols</u>. This option is useful for configuring connections across the Internet.

#### -PublisherLogin publisher\_login

Is the Publisher login name. If **PublisherSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

#### -PublisherNetwork publisher\_network

Is the Net-Library (without the .dll extension) to use when connecting to the Publisher. This option is useful when configuring the Merge Agent to connect to a Publisher over the Internet.

#### -PublisherPassword publisher\_password

Is the Publisher password. If **PublisherSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

#### -PublisherSecurityMode [0|1]

Specifies the security mode of the Publisher. A value of **0** indicates SQL Server Authentication (default), and a value of **1** indicates Windows Authentication Mode.

#### -QueryTimeOut query\_time\_out\_seconds

Is the number of seconds before the query times out. The default is 300 seconds.

#### -SubscriberDatabasePath subscriber\_database\_path
Is the path to the Jet database (.mdb file) if **SubscriberType** is **2** (allows a connection to a Jet database without an ODBC DSN).

## -SubscriberDBAddOption [0|1|2|3]

SubscriberDBAddOption	
value	Description
0	Use the existing database (default).
1	Create a new, empty Subscriber database.
2	Create a new database and attach it to the specified file.
3	Create a new database, attach the database, and enable all subscriptions that might exist at the file.

Specifies whether there is an existing Subscriber database.

**Note** When you use values **2** and **3**, the database path for the Subscriber must be specified in the **SubscriberDatabasePath** option.

-

#### SubscriberLogin subscriber\_login

Is the Subscriber login name. If **SubscriberSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

#### -SubscriberPassword subscriber\_password

Is the Subscriber password. If **SubscriberSecurityMode** is **0** (for SQL Server Authentication), this parameter must be specified.

## -SubscriberSecurityMode [0|1]

Specifies the security mode of the Subscriber. A value of **0** indicates SQL Server Authentication (default), and a value of **1** indicates Windows Authentication Mode.

# -SubscriberType [0|1|2|3|4|5|6|7|8]]

Specifies the type of Subscriber connection used by the Merge Agent.

SubscriberType value	Description
0	Microsoft SQL Server
1	ODBC data source
2	Jet database (direct connection)
3	OLE DB data source
4	Exchange data source
5	Oracle data source
6	DB2 data source
7	SSCE data source
8	XML data source

# -SubscriptionType [0|1|2]

Specifies the subscription type for distribution. A value of **0** indicates a push subscription (default), a value of **1** indicates a pull subscription, and a value of **2** indicates an anonymous subscription.

#### -**UploadGenerationsPerBatch** *upload\_generations\_per\_batch*

Is the number of generations to be processed in a single batch while uploading changes from the Subscriber to the Publisher. A generation is defined as a logical group of changes per article. The default for a reliable communication link is 100. The default for an unreliable communication link is 1.

#### -UploadReadChangesPerBatch upload\_read\_changes\_per\_batch

Is the number of changes to be read in a single batch while uploading changes from the Subscriber to the Publisher. The default is 100.

#### -UploadWriteChangesPerBatch upload\_write\_changes\_per\_batch

Is the number of changes to be applied in a single batch while uploading changes from the Subscriber to the Publisher. The default is 100.

#### -Validate [0|1|2|3]

Specifies whether validation should be done at the end of the merge session, and, if so, what type of validation.

Validate value	Description
0	No validation (default)
1	Rowcount-only validation
2	Rowcount and checksum validation
3	Rowcount and binary checksum
	validation

**Note** The value of **3** is the recommended value for use with SQL Server 2000; however, it cannot be used with earlier versions of SQL Server.

#### -ValidateInterval validate\_interval

Is how often, in minutes, the subscription is validated in continuous mode. The default is 60 minutes.

# Remarks

**IMPORTANT** If you have installed SQL Server Agent to run under a local system account rather than under a domain user account (the default), the service can access only the local computer. If the Merge Agent that runs under SQL Server Agent is configured to use Windows Authentication Mode when it logs in to SQL Server 2000, the Merge Agent fails. The default setting is SQL Server Authentication.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## See Also

Merge Agent Profile

**Replication Agents** 

# **Replication Queue Reader Agent Utility**

The Replication Queue Reader Agent utility configures and begins the Queue Reader Agent, which reads messages stored in a SQL Server queue or a Microsoft Message Queue and then applies those messages to the Publisher. Queue Reader Agent is used with snapshot and transactional publications that allow queued updating.

# Syntax

queueread [-?]
-Publisher server\_name[\instance\_name]
-PublisherDB publisher\_database
[-Distributor server\_name[\instance\_name]]

[-DistributionDB distribution\_database] [-DistributorLogin distributor\_login] [-DistributorPassword distributor\_password] [-DistributorSecurityMode [0|1]]

[-ResolverState [1|2|3]] [-HistoryVerboseLevel [1|2]]

[-PollingInterval polling\_interval]

[-Continuous] [-Output output\_path\_and\_file\_name] [-OutputVerboseLevel [0|1|2]]

[-LoginTimeOut login\_time\_out\_seconds]
[-QueryTimeOut query\_time\_out\_seconds]

[-DefinitionFile definition\_file]

**Note** Parameters can be specified in any order. When optional parameters are not specified, values from predefined registry settings on the local computer are used.

# Arguments

#### -?

Displays usage information.

#### -Publisher server\_name[\instance\_name]

Is the name of the Publisher. Specify *server\_name* for the default instance of Microsoft® SQL Server<sup>™</sup> 2000 on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server.

#### -PublisherDB publisher\_database

Is the name of the Publisher database.

#### -Distributor server\_name[\instance\_name]

Is the Distributor name. Specify *server\_name* for the default instance of SQL Server on that server. Specify *server\_name\instance\_name* for a named instance of SQL Server 2000 on that server. For Distributor (push) distribution, the name defaults to the name of the default instance of SQL Server on the local computer.

#### -DistributionDB distribution\_database

Is the distribution database.

#### -DistributorLogin distributor\_login

Is the Distributor login name.

#### -DistributorPassword distributor\_password

Is the Distributor password.

#### -DistributorSecurityMode [0|1]

Specifies the security mode of the Distributor. A value of **0** indicates SQL Server Authentication Mode (default), and a value of **1** indicates Windows Authentication Mode.

#### -ResolverState [1|2|3]

Specifies how conflicts will be resolved. A value of 1 indicates the Publisher wins the conflict and the subscription will be reinitialized. A value of 2 indicates

that the Publisher wins the conflict and the subscription will not be reinitialized. A value of 3 indicates the Subscriber wins the conflict.

#### -HistoryVerboseLevel [1|2]

Specifies the amount of history logged during a queue reader operation. You can minimize the performance affect of history logging by selecting **1**.

HistoryVerboseLevel	
value	Description
1	Default. Always update a previous history message of the same status (startup, progress, success, and so forth). If no previous record with the same status exists, insert a new record.
2	Insert new history records unless the record is for such things as idle messages or long-running job messages, in which case update the previous records.

#### -PollingInterval polling\_interval

Is how often, in seconds, the log is queried for replicated transactions. The default is 10 seconds. The value can be between 0 and 240 seconds.

#### -Continuous

Specifies whether the agent attempts to poll replicated transactions continually. If specified, the agent polls replicated transactions from the source at polling intervals even if there are no transactions pending.

#### -Output output\_path\_and\_file\_name

Is the path of the agent output file. If the file name is not provided, the output is sent to the console. If the specified file name exists, the output is appended to the file.

## -OutputVerboseLevel [0|1|2]

Specifies whether the output should be verbose. If the verbose level is **0**, only error messages are printed. If the verbose level is **1**, all the progress

report messages are printed. If the verbose level is **2** (default), all error messages and progress report messages are printed, which is useful for debugging.

#### -LoginTimeOut login\_time\_out\_seconds

Is the number of seconds before the login times out. The default is 15 seconds.

# -QueryTimeOut query\_time\_out\_seconds

Is the number of seconds before the query times out. The default is 300 seconds.

# -DefinitionFile def\_path\_and\_file\_name

Is the path of the agent definition file. An agent definition file contains command prompt arguments for the agent. The content of the file is parsed as an executable file. Double quotation marks (") can be used to specify argument values containing arbitrary characters.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# See Also

Queue Reader Agents

**Replication Agents** 

# **Replication Snapshot Agent Utility**

The Replication Snapshot Agent utility configures and begins the Snapshot Agent, which prepares snapshot files containing schema and data of published tables and database objects, stores the files in the snapshot folder, and records synchronization jobs in the distribution database.

# Syntax

snapshot [-?] -**Publisher** server name[\instance name] -PublisherDB publisher\_database -Publication publication\_name [-BcpBatchSize bcp batch size] [-Continuous] [**-DefinitionFile** *def\_path\_and\_file\_name*] [**-Distributor** server\_name[\instance\_name]] [**-DistributorLogin** *distributor\_login*] [**-DistributorPassword** *distributor\_password*] [-DistributorSecurityMode [0|1]] [-FieldDelimiter field\_delimiter] [-HistoryVerboseLevel [1|2|3] [-LoginTimeOut login\_time\_out\_seconds] [-MaxBcpThreads] [**-Output** *output\_path\_and\_file\_name*] [-OutputVerboseLevel [0|1|2] [-PublisherLogin publisher\_login] [-PublisherPassword publisher password] [-PublisherSecurityMode [0|1]] [-QueryTimeOut query\_time\_out\_seconds] [-ReplicationType [1|2]] [-RowDelimiter row\_delimiter]

Note Parameters can be specified in any order.

# Arguments

-?

Prints all available parameters.

#### -PublisherDB publisher\_database

Is the name of the Publisher database.

#### -Publication publication

Is the name of the publication. This parameter is only valid if the publication is set to always have a snapshot available for new or reinitialized subscriptions.

#### -BcpBatchSize bcp\_batch\_size

Is the number of rows to send in a bulk copy operation. When performing a **bcp in** operation, the batch size is the number of rows to send to the server as one transaction, and also the number of rows that must be sent before the Distribution Agent logs a **bcp** progress message. When performing a **bcp out** operation, a fixed batch size of 1000 is used. A value of 0 indicates no message logging.

#### -Continuous

Specifies whether the agent attempts to poll replicated transactions continually. If specified, the agent polls replicated transactions from the source at polling intervals even if there are no transactions pending.

#### -DefinitionFile def\_path\_and\_file\_name

Is the path of the agent definition file. An agent definition file contains command prompt arguments for the agent. The content of the file is parsed as an executable file. Double quotation marks (") can be used to specify argument values containing arbitrary characters.

#### -Distributor distributor

Is the Distributor name.

#### -DistributorLogin server\_name[\instance\_name]

Is the Distributor login name. Specify *server\_name* for the default instance of Microsoft® SQL Server<sup>™</sup> 2000 on that server. Specify *server\_name*\*instance\_name* for a named instance of SQL Server 2000 on

that server.

# -DistributorPassword distributor\_password

Is the Distributor password.

# -DistributorSecurityMode [0|1]

Specifies the security mode of the Distributor. A value of **0** indicates SQL Server Authentication Mode (default), and a value of **1** indicates Windows Authentication Mode.

# -FieldDelimiter field\_delimiter

Is the character or character sequence that marks the end of a field in the SQL Server bulk-copy data file. The default is n<x3>n.

# -HistoryVerboseLevel [1|2|3]

Specifies the amount of history logged during a snapshot operation. You can minimize the performance effect of history logging by selecting **1**.

HistoryVerboseLevel	
value	Description
1	Default. Always update a previous history message of the same status (startup, progress, success, and so forth). If no previous record with the same status exists, insert a new record.
2	Insert new history records unless the record is for such things as idle messages or long-running job messages, in which case update the previous records.
3	Always insert new records, unless it is for idle messages.

# -LoginTimeOut login\_time\_out\_seconds

Is the number of seconds before the login times out. The default is 15 seconds.

# -MaxBcpThreads

Specifies the number of bulk copy operations that can be performed in parallel. The maximum number of threads and ODBC connections that exist simultaneously is the lesser of **MaxBcpThreads** or the number of bulk copy requests that appear in the synchronization transaction in the distribution database. **MaxBcpThreads** must have a value greater than zero, and has no hard-coded upper limit. The default is 1.

## -Output output\_path\_and\_file\_name

Is the path of the agent output file. If the file name is not provided, the output is sent to the console. If the specified file name exists, the output is appended to the file.

## -OutputVerboseLevel [0|1|2]

Specifies whether the output should be verbose. If the verbose level is **0**, only error messages are printed. If the verbose level is **1**, all the progress report messages are printed. If the verbose level is **2** (default), all error messages and progress report messages are printed, which is useful for debugging.

## -PublisherLogin publisher\_login

Is the Publisher login name.

## -PublisherPassword publisher\_password

Is the Publisher password.

## -PublisherSecurityMode [0|1]

Specifies the security mode of the Publisher. A value of **0** indicates SQL Server Authentication (default), and a value of **1** indicates Windows Authentication Mode.

## -QueryTimeOut query\_time\_out\_seconds

Is the number of seconds before the query times out. The default is 30 seconds.

## -ReplicationType [1|2]

Specifies the type of replication. A value of **1** indicates transactional replication, and a value of **2** indicates merge replication.

#### -RowDelimiter row\_delimiter

Is the character or character sequence that marks the end of a row in the SQL Server bulk-copy data file. The default is  $n<,@g>\n$ .

# Remarks

**IMPORTANT** If you have installed SQL Server Agent to run under a Local System account rather than under a Domain User account (the default), the service can access only the local computer. If the Snapshot Agent that runs under SQL Server Agent is configured to use Windows Authentication Mode when it logs in to SQL Server, the Snapshot Agent fails. The default setting is SQL Server Authentication.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# See Also

Replication Agents
Snapshot Agent Profile

# scm Utility

The **scm** utility (the Service Control Manager) creates, modifies, starts, stops, or pauses any of the Microsoft® SQL Server<sup>™</sup> 2000 services that run under Microsoft Windows NT® and Microsoft Windows® 2000. Under Microsoft Windows 98, the **scm** utility starts, stops, or pauses the equivalent SQL Server applications.

# Syntax

scm [-?]
-Action {1 | 2 | 3 | 4 | 5 | 6 | 7}
-Service service\_name
[-Server server\_name]
[-Pwd sa\_password]
[-StartupOptions startup\_option [ ...n] ]
[-ExePath exe\_file\_path]
[-SvcStartType {1 | 2}]
[-SvcAccount service\_account]
[-SvcPwd service\_password]
[-Dependencies service\_name\_dependency [;...n] ]

# Arguments

# [-?]

Displays usage information.

# -Action $\{1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7\}$

Specifies which of these actions the utility performs.

Code	Action
1	Start
2	Restart
3	Is Running
4	Delete Service

5	Install Service
6	Stop
7	Modify

The **scm** utility returns a message box indicating if the action succeeded or failed. When using the **3** action code, success indicates the service is running and failure indicates the service is not running.

#### -Service service\_name

Specifies which SQL Server 2000 service is affected.

#### [-Server server\_name]

Specifies the server for which a service is affected. The default is the local computer.

**Note** Do not include an instance name with the *server\_name*. Only use the computer name. To affect a named instance, specify the unique *service\_name* with the **-Service** option.

#### [-Pwd sa\_password]

Is the **sa** login password on the server. The default is a blank password.

#### [-StartupOptions startup\_option [ ...n] ]

Specifies a space-delimited list of server startup options to be used when the service starts. This option is applicable when you install or modify the service.

## [-ExePath exe\_file\_path]

Specifies the file path to the service executable on the local computer. This option is applicable when you install or modify the service.

## $[-SvcStartType \ \{1 \ | \ 2\}]$

Specifies whether the service starts automatically. If **1** is specified, the service must be started manually. If **2** is specified, the service starts automatically when the computer starts. This option is applicable when you install or modify the service.

#### [-SvcAccount service\_account]

Specifies the network login account to assign to the service. This option is applicable when you install or modify the service.

[-SvcPwd service\_password]

Specifies the network login account password. This option is applicable when you install or modify the service.

#### [-Dependencies service\_name\_dependency [;...n]]

Specifies dependencies that this service has on other services. A dependent service can run only if the parent service is running. This option is applicable when you install or modify the service.

# Remarks

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# sqlagent Application

The **sqlagent** application starts SQL Server Agent from the command prompt. Usually, SQL Server Agent should be run from SQL Server Service Manager or by using SQL-DMO methods in an application. Only run **sqlagent** from the command prompt when you are diagnosing SQL Server Agent, or when you are directed to by your primary support provider.

# Syntax

sqlagent -c [-v]

# Arguments

-C

Indicates that SQL Server Agent is running from the command prompt and is independent of the Windows NT Service Control Manager. When **-c** is used, SQL Server Agent cannot be controlled from either the Services application in Control Panel or SQL Server Service Manager.

-V

Indicates that SQL Server Agent runs in verbose mode and writes diagnostic information to the command-prompt window. The diagnostic information is the same as the information written to the SQL Server Agent error log.

# Remarks

After displaying a copyright message, **sqlagent** displays output in the command prompt window only when the **-v** switch is specified. To stop **sqlagent**, press CTRL+C at the command prompt. Do not close the command-prompt window before stopping **sqlagent**.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# sqldiag Utility

The **sqldiag** utility gathers and stores diagnostic information and the contents of the query history trace (if running.) The output file includes error logs, output from **sp\_configure** and additional version information. If the query history trace was running when the utility was invoked, the trace file will contain the last 100 SQL events and exceptions. **sqldiag** is intended to expedite and simplify information gathering by Microsoft Product Support Services.

# Syntax

```
sqldiag [-?] |
  [-I instance_name]
  [ [-U login_ID] [-P password] | [-E] ]
  [-O output_file]
  [-X] [-M] [-C]
```

# Arguments

-?

Displays usage information.

-I instance\_name

Specifies the instance of Microsoft® SQL Server<sup>™</sup> 2000 on the local server on which to connect. Omit the **-I** option to connect to the default instance on the local server.

```
-U login_ID
```

Is the user login ID. Login IDs are case-sensitive.

```
-P password
```

Is the password for the specified *login\_ID*. If the **-P** option is specified at the end of the command prompt specifying *password*, **sqldiag** uses the default without password of NULL. Passwords are case-sensitive.

Uses a trusted connection instead of requesting a password.

-O output\_file

Redirects sqldiag output to the file named *output\_file*. If the **-O** option is not specified, the output file name defaults to sqldiag.txt. In this case, the trace file names remain unchanged as blackbox.trc and blackbox\_01.trc.

If the **-O** option is specified, sqldiag renames trace files blackbox.trc and blackbox\_01.trc based on the name used for *output\_file* (for example, if *output\_file* is specified as MyDiagnostics.txt, the trace files will be renamed to MyDiagnostics.trc and MyDiagnostics\_01.trc respectively).

Use of the **-O** option enables users to store several sqldiag outputs in the same directory.

# -X

Excludes error logs.

# -M

Performs DBCC stackdump.

## -C

Retrieves cluster information.

# Remarks

The **sqldiag** utility can be run anytime, whether or not SQL Server is started. If SQL Server is running, **sqldiag** gathers these items:

- Text of all error logs.
- Registry information.
- DLL version information.
- Output from:
  - sp\_configure

- sp\_who
- sp\_lock
- sp\_helpdb
- xp\_msver
- sp\_helpextendedproc
- sysprocesses
- Input buffer SPIDs/deadlock information.
- Microsoft Diagnostics Report for the server, including:
  - Contents of <servername>.txt file
  - Operating System Version Report
  - System Report
  - Processor List
  - Video Display Report
  - Hard Drive Report
  - Memory Report

- Services Report
- Drivers Report
- IRQ and Port Report
- DMA and Memory Report
- Environment Report
- Network Report
- The last 100 queries and exceptions.

If SQL Server 2000 is not running, **sqldiag** skips gathering the input buffer SPIDs information and the diagnostics report. **sqldiag** must be run on the server itself, not on a client workstation. If **sqldiag** is executed on a Microsoft Windows® 98 server, the **-U** and **-P** options are required, which means **sqldiag** must be run from a Microsoft MS-DOS® window. The information gathered by **sqldiag** for a Windows 98 server will be less complete than the information gathered on a Microsoft Windows 2000 or Microsoft Windows NT® server.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## See Also

**Reporting Errors to Your Primary Support Provider** 

# sqlmaint Utility

The **sqlmaint** utility performs a specified set of maintenance operations on one or more databases. Use **sqlmaint** to run DBCC checks, back up a database and its transaction log, update statistics, and rebuild indexes. All database maintenance activities generate a report that can be sent to a designated text file, HTML file, or e-mail account.

# Syntax

```
sqlmaint [-?] |
[-S server_name[\instance_name]]
  [-U login_ID [-P password]]
  {
    [-D database_name | -PlanName name | -PlanID guid ]
    [-Rpt text_file]
    [-To operator_name]
    [-HtmlRpt html_file [-DelHtmlRpt <time_period>]]
    [-RmUnusedSpace threshold_percent free_percent]
    [-CkDB | -CkDBNoIdx]
    [-CkAl | -CkAlNoIdx]
    [-CkCat]
    [-UpdOptiStats sample_percent]
    [-RebldIdx free_space]
    [-WriteHistory]
    ſ
       {-BkUpDB [backup path] | -BkUpLog [backup path] }
      {-BkUpMedia
         {DISK [ [-DelBkUps < time_period>]
               [-CrBkSubDir][-UseDefDir]
              1
         | TAPE
       }
      [-BkUpOnlyIfClean]
```

```
[-VrfyBackup]
]
}
```

#### <time\_period> ::= number[minutes | hours | days | weeks | months]

**Note** The parameters and their values must be separated by a space. For example, there must be a space between **-S** and *server*.

# Arguments

-?

Specifies that the syntax diagram for **sqlmaint** be returned. This parameter must be used alone.

```
-S server_name[\instance_name]
```

Specifies the target instance of Microsoft® SQL Server<sup>™</sup> 2000. Specify *server\_name* to connect to the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, **sqlmaint** connects to the default instance of SQL Server 2000 on the local computer.

```
-U login_ID
```

Specifies the login ID to use when connecting to the server. If not supplied, **sqlmaint** attempts to use Windows Authentication. If *login\_ID* contains special characters, it must be enclosed in double quotation marks ("); otherwise, the double quotation marks are optional.

```
-P password
```

Specifies the password for the login ID. Only valid if the **-U** parameter is also supplied. If *password* contains special characters, it must be enclosed in double quotation marks; otherwise, the double quotation marks are optional.

-D database\_name

Specifies the name of the database in which to perform the maintenance operation. If *database\_name* contains special characters, it must be enclosed

in double quotation marks; otherwise, the double quotation marks are optional.

#### -PlanName name

Specifies the name of a database maintenance plan defined using the Database Maintenance Plan Wizard. The only information **sqlmaint** uses from the plan is the list of the databases in the plan. Any maintenance activities you specify in the other **sqlmaint** parameters are applied to this list of databases. You can get the plan name from SQL Server Enterprise Manager.

#### -PlanID guid

Specifies the globally unique identifier (GUID) of a database maintenance plan defined using the Database Maintenance Plan Wizard. The only information **sqlmaint** uses from the plan is the list of the databases in the plan. Any maintenance activities you specify in the other **sqlmaint** parameters are applied to this list of databases. This must match a **plan\_id** value in **msdb.dbo.sysdbmaintplans**.

## -Rpt text\_file

Specifies the full path and name of the file into which the report is to be generated. The report is also generated on the screen. The report maintains version information by adding a date to the file name. The date is generated as follows: at the end of the file name but before the period, in the form \_yyyyMMddhhmm. Yyyy = year, MM = month, dd = day, hh = hour, mm = minute.

If you run the utility at 10:23 A.M. on December 1, 1996, and this is the *text\_file* value:

c:\Program Files\Microsoft SQL Server\Mssql\Backup\Nwind\_maint.r

The generated file name is:

c:\Program Files\Microsoft SQL Server\Mssql\Backup\Nwind\_maint\_1

The full UNC file name is required for *text\_file* when **sqlmaint** accesses a remote server.

#### -To operator\_name

Specifies the operator to whom the generated report will be sent through SQL Mail. The operator can be defined by using SQL Server Enterprise Manager.

#### -HtmlRpt html\_file

Specifies the full path and name of the file into which an HTML report is to be generated. **sqlmaint** generates the file name by appending a string of the format *\_yyyyMMddhhmm* to the file name, just as it does for the **-Rpt** parameter.

The full UNC file name is required for *html\_file* when **sqlmaint** accesses a remote server.

#### -DelHtmlRpt <time\_period>

Specifies that any HTML report in the report directory be deleted if the time interval after the creation of the report file exceeds *<time\_period>*.

-**DelHtmlRpt** looks for files whose name fits the pattern generated from the *html\_file* parameter. If *html\_file* is c:\Program Files\Microsoft SQL Server\Mssql\Backup\Nwind\_maint.htm, then -**DelHtmlRpt** causes **sqlmaint** to delete any files whose names match the pattern c:\Program Files\Microsoft SQL Server\Mssql\Backup\Nwind\_maint\*.htm and that are older than the specified <*time\_period*>.

#### -RmUnusedSpace threshold\_percent free\_percent

Specifies that unused space be removed from the database specified in **-D**. This option is only useful for databases that are defined to grow automatically. *Threshold\_percent* specifies in megabytes the size that the database must reach before **sqlmaint** attempts to remove unused data space. If the database is smaller than the *threshold\_percent*, no action is taken. *Free\_percent* specifies how much unused space must remain in the database, specified as a percentage of the final size of the database. For example, if a 200-MB database contains 100 MB of data, specifying 10 for *free\_percent* results in the final database size being 110 MB. Note that a database will not be expanded if it is smaller than *free\_percent* plus the amount of data in the database. For example, if a 108-MB database has 100 MB; it will remain at

108 MB.

# -CkDB | -CkDBNoIdx

Specifies that a DBCC CHECKDB statement or a DBCC CHECKDB statement with the NOINDEX option be run in the database specified in **-D**. For more information, see <u>DBCC CHECKDB</u>.

A warning is written to *text\_file* if the database is in use when **sqlmaint** runs.

# -CkAl | -CkAlNoIdx

Specifies that a DBCC NEWALLOC statement or a DBCC NEWALLOC statement with the NOINDEX option be run in the database specified in **-D**. For more information, see <u>DBCC NEWALLOC</u>.

## -CkCat

Specifies that a DBCC CHECKCATALOG statement be run in the database specified in **-D**. For more information, see <u>DBCC CHECKCATALOG</u>.

## -UpdOptiStats sample\_percent

Specifies that the following statement be run on each table in the database:

UPDATE STATISTICS table WITH SAMPLE sample\_percent PERCI

For more information, see <u>UPDATE STATISTICS</u>.

## -RebldIdx free\_space

Specifies that indexes on tables in the target database should be rebuilt by using the *free\_space* percent value as the inverse of the fill factor. For example, if *free\_space* percentage is 30, then the fill factor used is 70. If a *free\_space* percentage value of 100 is specified, then the indexes are rebuilt with the original fill factor value.

## -WriteHistory

Specifies that an entry be made in **msdb.dbo.sysdbmaintplan\_history** for each maintenance action performed by **sqlmaint**. If **-PlanName** or **-PlanID** is specified, the entries in **sysdbmaintplan\_history** use the ID of the specified plan. If **-D** is specified, the entries in **sysdbmaintplan\_history** are made with zeroes for the plan ID.

-BkUpDB [backup\_path] | -BkUpLog [backup\_path]

Specifies a backup action. **-BkUpDb** backs up the entire database. **- BkUpLog** backs up only the transaction log.

[*backup\_path*] specifies the directory for the backup. [*backup\_path*] is not needed if **-UseDefDir** is also specified, and is overriden by **-UseDefDir** if both are specified. The backup can be placed in a directory or a tape device address (for example, \\.\TAPE0). The file name for a database backup is generated automatically as follows:

# dbname\_db\_yyyyMMddhhmm.BAK

where

- *dbname* is the name of the database being backed up.
- *yyyyMMddhhmm* is the time of the backup operation with *yyyy* = year, *MM* = month, *dd* = day, *hh* = hour, and *mm* = minute.

The file name for a transaction backup is generated automatically with a similar format:

dbname\_log\_yyyymmddhhmm.BAK

If you use the **-BkUpDB** parameter, you must also specify the media by using the **-BkUpMedia** parameter.

# -BkUpMedia

Specifies the media type of the backup.

# DISK

Specifies that the backup medium is disk.

# -DelBkUps <time\_period>

Specifies that any backup file in the backup directory be deleted if the time interval after the creation of the backup exceeds the *<time\_period>*.

# -CrBkSubDir

Specifies that a subdirectory be created in the [*backup\_path*] directory or in the default backup directory if **-UseDefDir** is also specified. The name of the subdirectory is generated from the database name specified in **-D**. **- CrBkSubDir** offers an easy way to put all the backups for different databases into separate subdirectories without having to change the [*backup\_path*] parameter.

#### -UseDefDir

Specifies that the backup file be created in the default backup directory. **UseDefDir** overrides [*backup\_path*] if both are specified. With a default SQL Server 2000 setup, the default backup directory is c:\Program Files\Microsoft SQL Server\Mssql\Backup.

## TAPE

Specifies that the backup medium is tape.

# -BkUpOnlyIfClean

Specifies that the backup occur only if any specified **-Ck** checks did not find problems with the data. Maintenance actions run in the same sequence as they appear in the command prompt. Specify the parameters **-CkDB**, **-CkDBNoIdx**, **-CkAl**, **-CkAlNoIdx**, **-CkTxtAl**, or **-CkCat** before the **-BkUpDB/-BkUpLog** parameter(s) if you are also going to specify **-BkUpOnlyIfClean**, or the backup will occur whether or not the check reports problems.

## -VrfyBackup

Specifies that RESTORE VERIFYONLY be run on the backup when it completes.

#### number[minutes | hours | days | weeks | months]

Specifies the time interval used to determine if a report or backup file is old enough to be deleted. *number* is an integer. Valid examples are 12**weeks**, 3**months**, and 15**days**. If only *number* is specified, the default date part is **weeks**.

# Remarks

**sqlmaint** performs maintenance operations on one or more databases. If **-D** is specified, the operations specified in the remaining switches are performed only on the specified database. If **-PlanName** or **-PlanID** are specified, the only information **sqlmaint** retrieves from the specified maintenance plan is the list of databases in the plan. All operations specified in the remaining **sqlmaint** parameters are applied against each database in the list obtained from the plan. **sqlmaint** does not apply any of the maintenance activities defined in the plan itself.

The **sqlmaint** utility returns 0 if it runs successfully, or 1 if it fails. Failure is reported:

- If any of the maintenance actions fail.
- If **-CkDB**, **-CkDBNoIdx**, **-CkAl**, **-CkAlNoIdx**, **-CkTxtAl**, or **-CkCat** checks find problems with the data.
- If a general failure is encountered.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

# Examples

# A. Perform DBCC checks on the Northwind database

sqlmaint -S MyServer -U "sa" -P "SaPwd" -D Northwind -CkDB -CkA

# **B.** Update statistics using a 15% sample in all databases in a plan.

sqlmaint -S MyServer -U "sa" -P "SaPwd" -PlanName MyUserDBPlan

# C. Backup all the databases in a plan to their individual subdirecto

sqlmaint -S MyServer -U "sa" -P "SaPwd" -PlanName MyUserDBPlan

# sqlservr Application

The **sqlservr** application starts, stops, pauses, and continues an instance of Microsoft® SQL Server<sup>™</sup> 2000 from a command prompt.

# Syntax

```
sqlservr [-sinstance_name] [-c] [-dmaster_path] [-f]
[-eerror_log_path] [-lmaster_log_path] [-m]
[-n] [-Ttrace#] [-v] [-x] [-g number] [-O] [-y number]
```

# Arguments

-sinstance\_name

Specifies the instance of SQL Server to connect to. If no named instance is specified, **sqlservr** starts the default instance of SQL Server.

**IMPORTANT** When starting an instance of SQL Server, you must use the **sqlservr** application in the appropriate directory for that instance. For the default instance, run **sqlservr** from the \MSSQL\Binn directory. For a named instance, run **sqlservr** from the \MSSQL\$*instance\_name*\Binn directory.

-C

Indicates that an instance of SQL Server is started independently of the Windows NT Service Control Manager. This option is used when starting SQL Server from a command prompt, to shorten the amount of time it takes for SQL Server to start. (Note that when you use this option, you cannot stop SQL Server by using SQL Server Service Manager or the **net stop** command, and if you log off the Microsoft Windows NT® system, SQL Server will be stopped.)

## -dmaster\_path

Indicates the fully qualified path for the **master** database file. There are no spaces between **-d** and *master\_path*.

-f
Starts the server in minimally configured mode. The system administrator can then reconfigure configuration options (with the **sp\_configure** system stored procedure).

#### -eerror\_log\_path

Indicates the fully qualified path for the error log file. If not specified, the default location is *x*:\Program Files\Microsoft SQL Server\MSSQL\Log\Errorlog for the default instance and *x*:\Program Files\Microsoft SQL Server\MSSQL\$*instance\_name*\Log\Errorlog for a named instance. There are no spaces between **-e** and *error\_log\_path*.

### -lmaster\_log\_path

Indicates the fully qualified path for the **master** database transaction log file. There are no spaces between **-l** and *master\_log\_path*.

#### -m

Indicates to start an instance of SQL Server in single-user mode. Only a single user can connect when SQL Server is started in single-user mode. The CHECKPOINT mechanism, which guarantees that completed transactions are regularly written from the disk cache to the database device, is not started. (Typically, this option is used if you experience problems with system databases that require repair.)

#### -n

Indicates that you do not want to use the Windows NT application log to log SQL Server events. If you start an instance of SQL Server with the **-n** option, it is advisable to use the **-e** option too, or SQL Server events are not logged.

#### -Ttrace#

Indicates that an instance of SQL Server should be started with a specified trace flag (*trace#*) in effect. Trace flags are used to start servers with nonstandard behavior. For more information about available trace flags (*trace#*), see <u>Trace Flags</u>.

**IMPORTANT** When specifying a trace flag, use **-T** to pass the trace flag number. A lowercase t (**-t**) is accepted by SQL Server; however, **-t** sets other internal trace flags required by SQL Server support engineers.

-V

Displays the server version number.

-X

Disables maintaining CPU statistics.

-g memory\_to\_reserve

Specifies an integer number of megabytes of memory to reserve for other applications running within (in-process) SQL Server 2000.

-0

Specifies that Distributed COM (DCOM) is not required, thereby disabling heterogeneous queries.

-y error\_number

If SQL Server 2000 encounters an error message specified in this option, it writes the symptom stack trace to the error log. You can specify multiple errors by using multiple **–y** arguments.

## Remarks

For more information about where to find or how to run this utility, see <u>Getting</u> <u>Started with Command Prompt Utilities</u>.

Command Prompt Utilities

# sqlftwiz Utility

The **sqlftwiz** utility allows the Full-Text Indexing Wizard to be executed using a command prompt utility. The Full-Text Indexing Wizard can also be started from SQL Server Enterprise Manager. Use the Full-Text Indexing Wizard to define full-text indexing on Microsoft® SQL Server<sup>™</sup> 2000 text-based columns with either an existing full-text catalog, or a new full-text catalog. The Full-Text Indexing Wizard also creates or modifies population schedules that determine when the information stored in the full-text catalog is updated.

## Syntax

sqlftwiz {/n | {/ulogin\_id /ppassword}}
[/sserver\_name[\instance\_name]]
[/ddatabase\_name]

# Arguments

/n

Specifies Windows Authentication Mode (password not required).

### /**u**login\_id

Specifies the login ID used to connect to an instance of SQL Server 2000.

/**p**password

Specifies the password for the given *login\_id*.

### /sserver\_name[\instance\_name]

Specifies the instance of SQL Server 2000 whose tables are to be full-text indexed. Specify *server\_name* to connect to the default instance of SQL Server 2000 on that server. Specify *server\_name\instance\_name* to connect to a named instance of SQL Server 2000 on that server. If no server is specified, **sqlftwiz** connects to the default instance of SQL Server 2000 on the local computer.

/**d**database\_name

Specifies the database name of the SQL Server database whose tables are to be full-text indexed. If *database\_name* is not specified, SQL Server 2000 prompts for the selection of a *database\_name* on the given *server\_name*.

## Remarks

There must be no space between a switch and its corresponding value.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.

## Examples

This example runs the **sqlftwiz** utility against the **Northwind** database on a server named **MyServer**.

sqlftwiz /sMyServer /n /dNorthwind

## See Also

<u>CONTAINS</u>

FREETEXT

Full-Text Querying SQL Server Data

Command Prompt Utilities

# vswitch Utility

The **vswitch** utility can be used to switch between Microsoft® SQL Server<sup>™</sup> 2000, SQL Server version 6.5, and SQL Server version 6.0 as the active version of SQL Server.

## Syntax

vswitch -SwitchTo {60 | 65 | 80} [-Silent {0 | 1}]

## Arguments

#### -SwitchTo {60 | 65 | 80}

Specifies the version of SQL Server to make active.

### -Silent $\{0 \mid 1\}$

Specifies whether any user interface or messages are displayed. If **1**, no user interface or messages are displayed. The default is **0**.

## Remarks

The **vswitch** utility affects only the default instance of SQL Server. It does not affect named instances of SQL Server 2000. For more information, see <u>Multiple</u> <u>Instances of SQL Server</u>.

For information about where to find or how to run this utility, see <u>Getting Started</u> <u>with Command Prompt Utilities</u>.