

CHAPTER 14

Utilities and Options Reference

About this chapter This chapter provides reference material for the SQL Remote command line utilities and SQL Remote database options.

Contents

Topic	Page
The Message Agent	306
The Database Extraction utility	314
The SQL Remote Open Server	321
SQL Remote options	323

The Message Agent

Purpose To send and apply SQL Remote messages, and to maintain the message tracking system to ensure message delivery.

Syntax { **dbremote** | **ssremote** } [*switches*] [*directory*]

Command-line switches

Switch	Description
<i>@filename</i>	Read in switches from configuration file
<i>@envvar</i>	Read in switches from environment variable
-a	Do not apply received transactions
-b	Run in batch mode
-c " <i>keyword=value; ...</i> "	Supply database connection parameters
-cq " <i>keyword=value; ...</i> "	Supply database connection parameters for the stable queue (Adaptive Server Enterprise only)
-dl	Display log messages on screen
-e <i>locale-string</i>	Locale setting (Adaptive Server Enterprise only)
-fq	Full scan of the stable queue when sending messages (Adaptive Server Enterprise only)
-g <i>n</i>	Group transactions consisting of less than <i>n</i> operations.
-i	Scan transactions from the transaction log into the stable queue (Adaptive Server Enterprise only).
-k	Close window on completion
-l <i>length</i>	Maximum message length
-m <i>size</i>	Maximum amount of memory used for building messages.
-o <i>file</i>	Output messages to file
-ot <i>file</i>	Truncate file and log output messages
-p	Do not purge messages
-q	Run with minimized window
-r	Receive messages
-rd <i>minutes</i>	Polling frequency for incoming messages
-rp <i>number</i>	Number of receive polls before message is assumed lost
-ru <i>time</i>	Waiting period to re-scan log on receipt of a resend.
-s	Send messages

-sd <i>time</i>	Send polling period
-t	Replicate all triggers (Adaptive Server Anywhere only)
-u	Process only backed up transactions
-v	Verbose operation
-w <i>n</i>	Number of worker threads to apply incoming messages (Windows NT and Solaris only)
-x	Rename and restart the transaction log (Adaptive Server Anywhere only).
directory	The directory in which old transaction logs are held (Adaptive Server Anywhere only)

Description

The Message Agent sends and applies messages for SQL Remote replication, and maintains the message tracking system to ensure message delivery.

The name of the Message Agent executable is as follows:

- ◆ **dbremote** The Message Agent for Adaptive Server Anywhere.
- ◆ **ssremote** The Message Agent for Adaptive Server Enterprise.

For Adaptive Server Anywhere, the user ID in the Message Agent command line must have either REMOTE DBA or DBA authority. For Adaptive Server Enterprise, the user ID must have replication role.

The optional *directory* parameter specifies a directory in which old transaction logs are held, so that the Message Agent has access to events from before the current log was started.

The Message Agent uses a number of connections to the database. For a listing, see "Connections used by the Message Agent" on page 240.

For information on REMOTE DBA authority, see "The Message Agent and replication security" on page 257.

Command-line switch details

@filename Read in command-line switches from the supplied file.

The file may contain line breaks, and may contain any set of command line switches. For example, the following command file holds a set of command line switches for a Message Agent that starts with a cache size of 4 Mb, sends messages only, and connects to a database named **field** on a server named **myserver**:

```
-m 4096
-s
-c "eng=myserver;dbn=field;uid=sa;pwd=sysadmin"
```

If this configuration file is saved as *c:\config.txt*, it can be used in an command line as follows:

```
ssremote @c:\config.txt
```

or

```
dbremote @c:\config.txt
```

@environment-variable Read in command-line switches from the supplied environment variable.

The environment variable may contain any set of command line switches. For example, the first of the following pair of statements sets an environment variable holding a set of command line switches for a database server that starts with a cache size of 4 Mb, receives messages only, and connects to a database named **field** on a server named **myserver**. The **set** statement should be entered all on one line:

```
set envvar=-m 4096 -s -c  
"eng=myserver;dbn=field;uid=sa;pwd=sysadmin"  
ssremote @envvar
```

-a Process the received messages (those in the inbox) without applying them to the database. Used together with **-v** (for verbose output) and **-p** (so the messages are not purged), this flag can help detect problems with incoming messages. Used without **-p**, this flag purges the inbox without applying the messages, which may be useful if a subscription is being restarted.


-b Run in batch mode. In this mode, the Message Agent processes incoming messages, scans the transaction log once and processes outgoing messages, and then stops.

-c "parameter=value; ..." Specify connection parameters. For Adaptive Server Anywhere, if this option is not specified, the environment variable SQLCONNECT is used.

For example, the following statement runs *dbremote* on a database file named *c:\asa6\asademo.db*, connecting with user ID **DBA** and password **SQL**:

```
dbremote -c "uid=dba;pwd=sql;dbf=c:\asa6\asademo.db"
```

The Message Agent must be run by a user with REMOTE DBA authority or DBA authority.

 For information on REMOTE DBA authority, see "The Message Agent and replication security" on page 257.

The Message Agent for Adaptive Server Anywhere supports the full range of Adaptive Server Anywhere connection parameters. The Message Agent for Adaptive Server Enterprise supports the following connection parameters:

Parameter	Description
UID	Login ID
PWD	Password
DBN	(optional) Database name. If this parameter is not supplied, the connection defaults to the default database for the login ID.
ENG	Adaptive Server Enterprise name.

-cq "parameter=value; ..." Specify connection parameters for the stable queue. This option applies to Adaptive Server Enterprise only. If not supplied, the values default to the `-c` values.

-dl Display messages in the Message Agent window or on the command line and also in the log file.

-e locale-string This option applies to Adaptive Server Enterprise only. Specify Adaptive Server Enterprise locale information. The locale string has the following format:

```
"language_name, charset_name [, sort_order]"
```

By default, the Message Agent uses the default locale, which is defined in the file `sybase\locales\locales.dat`.

If `language_name` and `charset_name` are not supplied, the Message Agent obtains them from Adaptive Server Enterprise. If `sort_order` is not supplied, the Message Agent uses a binary sort order (sort by byte value).


-fq This option is for use only with Adaptive Server Enterprise. It permits a full scan of the stable queue when sending messages, starting from the oldest `confirm_sent` value.

This feature is intended for occasional use to clean out a large stable queue. If, for example, a single user has not confirmed receipt of a message from a long time ago, the stable queue may be very large. However, by running `-fq` you can delete entries from more up-to-date users that have been confirmed, even though they are more recent than the cutoff value at which entries are deleted by default.

-g n Instructs the Message Agent to group transactions containing less than `n` operations together with transactions that follow. The default is twenty operations. Increasing the value of `n` can speed up processing of incoming messages, by doing less commits. However, it can also cause deadlock and blocking by increasing the size of transactions.

-i Scan transactions from the transaction log into the stable queue. This option is available for Adaptive Server Enterprise only. It is used when you wish to run a separate copy of the Message Agent for scanning the transaction log and for sending and receiving messages.

If none of `-r`, `-i`, or `-s` is specified, the Message Agent executes all three phases. Otherwise, only the indicated phases are executed.

 For more information, see "Running multiple Message Agents" on page 283.

-k Close window on completion.

-l length Specifies the maximum length of message to be sent, in bytes. Longer transactions are split into more than one message. The default is 51200 (50K).

Caution

The maximum message length must be the same at all sites in an installation.

For platforms with restricted memory allocation, the value must be less than the maximum memory allocation of the operating system. For example, Windows 3.x allows a maximum memory allocation of 64K, and the maximum message length must be less than this..

-m size Specifies a maximum amount of memory to be used by the Message Agent for building messages and caching incoming messages. The allowed size can be specified as *n* (in bytes), *nK*, or *nM*. The default is 2048K (2M).

When all remote databases are receiving unique subsets of the operations being replicated, a separate message for each remote database is built up concurrently. Only one message is built for a group of remote users that are receiving the same operations. When the memory being used exceeds the `-m` value, messages are sent before reaching their maximum size (as specified by the `-l` switch).

When messages arrive, they are stored in memory by the Message Agent until they are applied. This caching of messages prevents rereading of that are out of order messages from the message system, which may lower performance on large installations. When the memory usage specified using the `-m` switch is exceeded, messages are flushed in a least-recently used fashion.

-o Append output to a log file. Default is to send output to the screen.

-ot Truncate the log file and then append output messages to it. Default is to send output to the screen.

-p Process the messages without purging them.

-q For Windowing operating systems only, starts the Message Agent with a minimized window.

-r Receive messages. If none of `-r`, `-i`, or `-s` is specified, the Message Agent executes all three phases. Otherwise, only the indicated phases are executed.

The Message Agent runs in continuous mode if called with `-r`. To have the Message Agent shut down after receiving messages, use the `-b` switch in addition to `-r`.

-rd time By default, the Message Agent polls for incoming messages every minute. This option (`rd` stands for **receive delay**) allows the polling frequency to be configured, which is useful when polling is expensive. The Message Agent checks for incoming messages once after each send cycle, regardless of the polling frequency, so if a value greater than the SEND EVERY frequency is supplied, the Message Agent checks for incoming messages once after each send cycle.

You can use a suffix of `s` after the number to indicate seconds, which may be useful if you want frequent polling. For example:

```
dbremote -f 30s
```

polls every thirty seconds.

☞ For more information on polling, see "Tuning incoming message polling" on page 245.

-rp When running in continuous mode, the Message Agent polls at certain intervals for messages. After polling a set number of times (by default, one), if a message is missing, the Message Agent assumes it has got lost and requests that it be resent. On slow message systems, this can result in many unnecessary resend requests. You can set the number of polls before a resend request is issued using this option, to cut down on the number of resend requests.

☞ For more information on configuring this option, see "Tuning incoming message polling" on page 245.

-ru Control the **resend urgency**. This is the time between detection of a resend request and when the Message Agent starts fulfilling the request. Use this switch to help the Message Agent collect resend requests from multiple users before rescanning the log. The time unit can be any of {s = seconds; m = minutes; h = hours; d = days}

-s Send messages. If none of `-r`, `-i`, or `-s` is specified, the Message Agent executes all three phases. Otherwise, only the indicated phases are executed.

-sd time Control the **send delay**, which is the time to wait between polls for more transaction log data to send.

-t All trigger actions are replicated. If you do use this switch, you must ensure that the trigger actions are not carried out twice at remote databases, once by the trigger being fired at the remote site, and once by the explicit application of the replicated actions from the consolidated database.

To ensure that trigger actions are not carried out twice, you can wrap an IF CURRENT REMOTE USER IS NULL ... END IF statement around the body of the triggers.

-u Process only transactions that have been backed up. This switch prevents the Message Agent from processing transactions since the latest backup. Using this switch, outgoing transactions and confirmation of incoming transactions are not sent until they have been backed up.

In Adaptive Server Anywhere, this means only transactions from renamed logs are processed. In Adaptive Server Enterprise, this means that only transactions committed before the latest **dump database** or **dump transaction** statement are processed.

-v Verbose output. This switch displays the SQL statements contained in the messages to the screen and, if the `-o` switch is used, to a log file.

-w n Windows NT and Solaris only. The number of worker threads used to apply incoming messages. The default is zero, which means all messages are applied by the main (and only) thread. A value of 1 (one) would have one thread receiving messages from the message system and one thread applying messages to the database.

The `-w` switch makes it possible to increase the throughput of incoming messages with hardware upgrades. Putting the consolidated database on a device that can perform many concurrent operations (a RAID array with a striped logical drive) will improve throughput of incoming messages. Multiple processors in the computer running the Message Agent could also improve throughput of incoming messages.

The `-w` switch will not improve performance significantly on hardware that cannot perform many concurrent operations.

Incoming messages from a single remote database will never be applied on multiple threads. Messages from a single remote database are always applied serially in the correct order.

**Message system
control parameters**

-x Rename and restart the transaction log after it has been scanned for outgoing messages. In some circumstances, replicating data to a consolidated database can take the place of backing up remote databases, or renaming the transaction log when the engine is shut down.

SQL Remote uses several registry settings (in Windows 95 and NT) or initialization file settings (Windows 3.x) to control aspects of message link behavior.

The message link control parameters are stored in the following places:

- ◆ **Windows 95 and Windows NT** In the registry, at the following location:

```
\\HKEY_CURRENT_USER
  \Software
    \Sybase
      \SQL Remote
```

- ◆ **Windows 3.x** In the file SQLANY.INI, in your SQL Remote installation directory.
- ◆ **NetWare** You should create a file named *dbremote.ini* in the *sys:\system* directory to hold the FILE system directory setting.

☞ For a listing of registry settings, see the section for each message system under "Using message types" on page 228.

The Database Extraction utility

You can access the remote database extraction utility in the following ways:

- ◆ From Sybase Central, for interactive use under Windows 95 or NT.
- ◆ From the system command line, using the *ssextract* or *dbextract* command-line utilities. This is useful for incorporating into batch or command files.

ssextract is the extraction utility for Adaptive Server Enterprise, *dbextract* is the extraction utility for Adaptive Server Anywhere.

By default, the extraction utility runs at isolation level zero. If you are extracting a database from an active server, you should run it at isolation level 3 (see "Extraction utility options" on page 317) to ensure that data in the extracted database is consistent with data on the server. Running at isolation level 3 may hamper others' turnaround time on the server because of the large number of locks required. It is recommended that you run the extraction utility when the server is not busy, or run it against a copy of the database (see "Designing an efficient extraction procedure" on page 211).

Objects owned by **dbo**

The **dbo** user ID owns a set of Adaptive Server Enterprise-compatible system objects in an Adaptive Server Anywhere database.

For Adaptive Server Anywhere, the extraction utility does not unload the objects created for the **dbo** user ID during database creation. Changes made to these objects, such as redefining a system procedure, are lost when the data is unloaded. Any objects created by the **dbo** user ID since the initialization of the database are unloaded by the Extraction utility, and so these objects are preserved.

Adaptive Server Enterprise security issue


If you extract an Adaptive Server Enterprise database, the resulting Adaptive Server Anywhere database has the default **DBA** user ID present. If you do not want this user ID present in the remote database, you need to explicitly REVOKE CONNECT for this user.

Extracting a remote database in Sybase Central

Running the extraction utility from Sybase Central carries out the following tasks related to creating and synchronizing SQL Remote subscriptions:

- ◆ Creates a command file to build a remote database containing a copy of the data in a specified publication.

- ◆ Creates the necessary SQL Remote objects, such as message types, publisher and remote user IDs, publication and subscription, for the remote database to receive messages from and send messages to the consolidated database.
 - ◆ Starts the subscription at both the consolidated and remote databases.
- ❖ **To extract a remote database from a running database:**
- 1 Connect to the database.
 - 2 Right-click the database and click Extract Database in the popup menu.
 - 3 Follow the instructions in the wizard.
- ❖ **To extract a remote database from a database file or a running database as follows:**
- 1 Open the Database Utilities folder in the left panel.
 - 2 Double-click Extract a Database in the right panel.
 - 3 Follow the instructions in the wizard.

 For full information on extracting a remote database in Sybase Central, see the Sybase Central online Help.

The extraction command-line utility

Purpose To extract a remote Adaptive Server Anywhere database from a consolidated Adaptive Server Enterprise or Adaptive Server Anywhere database.

Syntax { **SSXTRACT** | **DBXTRACT** } [*switches*] [*directory*] *subscriber*

Switch	Description
-an <i>database</i>	Creates a database file with the same settings as the database being unloaded and automatically reloads it.
-ac " <i>keyword=value; ...</i> "	Connect to the database specified in the connect string to do the reload.
-b	Do not start subscriptions
-c " <i>keyword=value; ...</i> "	Supply database connection parameters
-d	Unload data only
-f	Extract fully qualified publications
-l <i>level</i>	Perform all extraction operations at specified isolation level
-k	Close window on completion

-n	Extract schema definition only
-o file	Output messages to file
-p character	Escape character
-q	Operate quietly: do not print messages or show windows
-r file	Specify name of generated reload Interactive SQL command file (default "reload.sql")
-u	Unordered data
-v	Verbose messages
-x	Use external table loads
-xf	Exclude foreign keys
-xp	Exclude stored procedures
-xt	Exclude triggers
-xv	Exclude views
-y	Overwrite command file without confirmation
directory	The directory to which the files are written. This is not needed if you use <code>-an</code> or <code>-ac</code>
subscriber	The subscriber for whom the database is to be extracted.

Description

ssxtract is the extraction utility for Adaptive Server Enterprise. It is run against a Adaptive Server Enterprise and creates a command file for a remote Adaptive Server Anywhere database.

dbxtract is the extraction utility for Adaptive Server Anywhere. It is run against an Adaptive Server Anywhere database and creates a command file for a remote Adaptive Server Anywhere database.

There is no extraction utility to create remote Adaptive Server Enterprise databases.

The command line extraction utility creates a command file and a set of associated data files. The command file can be run against a newly-initialized Adaptive Server Anywhere database to create the database objects and load the data for the remote database.

By default, the command file is named *reload.sql*.

SSXtract notes

Not all Adaptive Server Enterprise objects have corresponding objects in Adaptive Server Anywhere. The *ssxtract* utility has the following limitations:

- ◆ **Single database** All extracted objects must be in a single Adaptive Server Enterprise database.

- ◆ **Passwords** The password for the extracted user IDs are the same as the user ID itself.
- ◆ **Permissions** The extracted user ID is granted REMOTE DBA authority.
- ◆ **Named constraints** These are extracted as Adaptive Server Anywhere CHECK constraints.
- ◆ **System tables** The `sp_populate_sql_anywhere` SQL Remote procedure builds a set of Adaptive Server Anywhere system tables in TEMPDB from the Adaptive Server Enterprise system tables. The extracted schema comes from these temporary system tables.

☞ For more information about the command-line switches, see "Extraction utility options" on page 317.

Extraction utility options

Create a database for reloading (-an) You can combine the operations of unloading a database, creating a new database, and loading the data using this option.

For example, the following command (which should be entered all on one line) creates a new database file named *asacopy.db* and copies the schema and data for the **field_user** subscriber of *asademo.db* into it:

```
dbxtract -c "uid=dba;pwd=sql;dbf=asademo.db" -an
asacopy.db field_user
```

If you use this option, no copy of the data is created on disk, so you do not specify an unload directory on the command line. This provides greater security for your data, but at some cost for performance.

Reload the data to an existing database (-ac) You can combine the operation of unloading a database and reloading the results into an existing database using this option.

For example, the following command (which should be entered all on one line) loads a copy of the data for the **field_user** subscriber into an existing database file named *newdemo.db*:

```
dbxtract -c "uid=dba;pwd=sql;dbf=asademo.db" -ac
"uid=dba;pwd=sql;dbf=newdemo.db" field_user
```

If you use this option, no copy of the data is created on disk, so you do not specify an unload directory on the command line. This provides greater security for your data, but at some cost for performance.

Do not start subscriptions automatically (-b) If this option is selected, subscriptions at the consolidated database (for the remote database) and at the remote database (for the consolidated database) must be started explicitly using the START SUBSCRIPTION statement for replication to begin.

Connection parameters (-c) A set of connection parameters, in a string.

- ◆ **DBXTRACT connection parameters** The **user ID** should have DBA authority to ensure that the user has permissions on all the tables in the database.

For example, the following statement (which should be typed on one line) extracts a database for remote user ID **joe_remote** from the *asademo* database running on the **sample_server** server, connecting as user ID DBA with password SQL. The data is unloaded into the *c:\unload* directory.

```
SSXTRACT -c "eng=sample_server;dbn=sademo;
uid=dba;pwd=sql" c:\extract joe_remote
```

If connection parameters are not specified, connection parameters from the SQLCONNECT environment variable are used, if set.

- ◆ **SSXTRACT connection parameters** The following connection parameters are supported:

Parameter	Description
UID	Login ID
PWD	Password
DBN	(optional) Database name. If this parameter is not supplied, the connection defaults to the default database for the login ID.
ENG	Adaptive Server Enterprise name.

ssxtract cannot extract passwords. It sets passwords to be the same as the user ID.

Unload the data only (-d) If this option is selected, the schema definition is not unloaded, and publications and subscriptions are not created at the remote database. This option is for use when a remote database already exists with the proper schema, and needs only to be filled with data.

Extract fully qualified publications (-f) In most cases, you do not need to extract fully qualified publication definitions for the remote database, since it typically replicates all rows back to the consolidated database anyway.

However, you may want fully qualified publications for multi-tier setups or for setups where the remote database has rows that are not in the consolidated database.

Perform extraction at a specified isolation level (-l) The default setting is an isolation level of zero. If you are extracting a database from an active server, you should run it at isolation level 3 (see "Extraction utility options" on page 317) to ensure that data in the extracted database is consistent with data on the server. Increasing the isolation level may result in large numbers of locks being used by the extraction utility, and may restrict database use by other users.

Unload the schema definition only (-n) With this definition, none of the data is unloaded. The reload file contains SQL statements to build the database structure only. You can use the SYNCHRONIZE SUBSCRIPTION statement to load the data over the messaging system. Publications, subscriptions, PUBLISH and SUBSCRIBE permissions are part of the schema.

Output messages to file (-o) Outputs the messages from the extraction process to a file for later review.

Escape character (-p) The default escape character (\) can be replaced by another character using this option.

Operate quietly (-q) Display no messages except errors. This option is not available from other environments. This is available only from the command-line utility.

Reload filename (-r) The default name for the reload command file is *reload.sql* in the current directory

Output the data unordered (-u) By default the data in each table is ordered by primary key. Unloads are quicker with the `-u` switch, but loading the data into the remote database is slower.

Verbose mode (-v) The name of the table being unloaded and the number of rows unloaded are displayed. The SELECT statement used is also displayed.

Use external loads (-x) In the reload script, the default is to use the LOAD TABLE statement to load the data into the database. If you choose to use external loads, the Interactive SQL INPUT statement is used instead. The LOAD TABLE statement is faster than INPUT.

INPUT takes the path of the data files relative to the client, while LOAD TABLE takes the path relative to the server.

Exclude foreign key definitions (-xf) You can use this if the remote database contains a subset of the consolidated database schema, and some foreign key references are not present in the remote database.

Exclude stored procedure (-xp) Do not extract stored procedures from the database.

Exclude triggers (-xt) Do not extract triggers from the database.

Exclude views (-xv) Do not extract views from the database.

Operate without confirming actions (-y) Without this option, you are prompted to confirm the replacement of an existing command file.

The SQL Remote Open Server

Purpose To take replication data from Replication Server and apply it to the SQL Remote stable queue. This utility is needed only for databases participating in both Replication Server (and using a Replication Agent) and SQL Remote replication.

Syntax **ssqueue** [*switches*] [*open-server-name*]

Command-line switches

Switch	Description
<i>open-server-name</i>	An open server name, which must be declared in the interfaces file (sql.ini).
-c " <i>keyword=value; ...</i> "	Supply database connection parameters
-cq " <i>keyword=value; ...</i> "	Supply database connection parameters for the stable queue
-dl	Display messages in window
-o <i>file</i>	Output messages to file
-v	Verbose operation

Description The SQL Remote Open Server is used to enable an Adaptive Server Enterprise database to take part in both SQL Remote replication while acting as a primary site in a Replication Server installation (or a replicate site using asynchronous procedure calls).

The name of the executable is as follows:

- ◆ **ssqueue.exe** Windows NT platforms.
- ◆ **ssqueue** UNIX platforms.

Command-line switch details

open-server-name Replication Server must connect to the SQL Remote Open Server, which therefore must have an open server name. This open server name is set on the command line, and must correspond to a master and query entry in the interfaces file (*sql.ini*) on the machine running the SQL Remote Open Server, and to a query entry on the interfaces file of the machine running Replication Server.

The default value for the open server name is **SSQueue**.

-c "parameter=value; ..." Specify connection parameters to the database holding the data being replicated. This connection is required for the SQL Remote Open Server to gain access to the SQL Remote system tables.

The connection parameters must come from the following list:

Parameter	Description
UID	Login ID
PWD	Password
DBN	(optional) Database name. If this parameter is not supplied, the connection defaults to the default database for the login ID.
ENG	Server name.

-cq "parameter=value; ..." Specify connection parameters for the stable queue. If not supplied, the values default to the `-c` values.

-dl Display messages in the window or on the command line and also in the log file.

-o Append output to a log file. Default is to send output to the screen.

-v Verbose output. This switch displays the SQL statements contained in the messages to the screen and, if the `-o` switch is used, to a log file.

SQL Remote options

Function Replication options are database options included to provide control over replication behavior.

Anywhere Syntax **SET [TEMPORARY] OPTION**
... [*userid.* | **PUBLIC.**] *option_name* = [*option_value*]

Enterprise syntax: **exec sp_remote_option** *option-name*, *option-value*

Argument	Description
<i>option_name</i>	The name of the option being changed.
<i>option-value</i>	A string containing the setting for the option.

Description The following options are available.

OPTION	VALUES	DEFAULT
Blob_threshold	Integer, in K	256
Compression	-1 to 9	6
Delete_old_logs	ON, OFF	OFF
Qualify_owners	ON, OFF	OFF
Quote_all_identifiers	ON, OFF	OFF
Replication_error	<i>procedure-name</i>	NULL
Subscribe_by_remote	ON,OFF	ON
Verify_threshold	<i>integer</i>	256
Verify_all_columns	ON,OFF	OFF

These options are used by the Message Agent, and should be set for the user ID specified on the Message Agent command line. They can also be set for general public use.

The options are as follows:

- ◆ **Blob_threshold option** Any value longer than the Blob_threshold option is replicated as a blob. That is, it is broken into pieces and replicated in chunks, before being reconstituted by using a SQL variable and concatenating the pieces at the recipient site.

If you are replicating blobs in an installation with Adaptive Server Enterprise, you must ensure that Blob_threshold is set to a value larger than the largest blob being replicated.

For information on blob replication and Adaptive Server Enterprise, see "Replication of blobs" on page 104.

- ◆ **Compression option** Set the level of compression for messages. Values can be from -1 to 9, and have the following meanings:
 - ◆ **-1** Send messages in Version 5 format. Message Agents (both *dbremote* and *ssremote*) from previous versions of SQL Remote cannot read messages sent in Version 6 format. You should ensure that **COMPRESSION** is set to -1 until all Message Agents in your system are upgraded to Version 6.
 - ◆ **0** No compression.
 - ◆ **1 to 9** Increasing degrees of compression. Creating messages with high compression can take longer than creating messages with low compression.
- ◆ **Delete_old_logs option** This option is used by SQL Remote and by the Adaptive Server Anywhere Replication Agent. The default setting is OFF. When set to ON, the Message Agent (DBREMOTE) deletes each old transaction log when all the changes it contains have been sent and confirmed as received.
- ◆ **Qualify_owners option** Controls whether SQL statements being replicated by SQL Remote should use qualified object names. The default in Adaptive Server Anywhere is ON and the default in Adaptive Server Enterprise is OFF.

Qualifying owners in Adaptive Server Enterprise setups is rarely needed because it is common for objects to be owned by **dbo**. When qualification is not needed in Adaptive Server Anywhere setups, messages will be slightly smaller with the option off.

- ◆ **Quote_all_identifiers option** Controls whether SQL statements being replicated by SQL Remote should use quoted identifiers. The default is OFF.

When this option is off, the *dbremote* quotes identifiers that require quotes by Adaptive Server Anywhere (as it has always done) and *ssremote* does not quote any identifiers. When the option is on, all identifiers are quoted.

- ◆ **Replication_error option** Specifies a stored procedure called by the Message Agent when a SQL error occurs. By default no procedure is called.

The replication error procedure must have a single argument of type CHAR, VARCHAR, or LONG VARCHAR. The procedure is called once with the SQL error message and once with the SQL statement that causes the error.

While the option allows you to track and monitor SQL errors in replication, you must still design them out of your setup: this option is not intended to resolve such errors.

- ◆ **Subscribe_by_remote option** When set to ON, operations from remote databases on rows with a subscribe by value that is NULL or an empty string assume the remote user is subscribed to the row. When set to OFF, the remote user is assumed not to be subscribed to the row.

The only limitation of this option is that it will lead to errors if a remote user really does want to INSERT (or UPDATE) a row with a NULL or empty subscription expression (for information held only at the consolidated database). This is reasonably obscure and can be worked around by assigning a subscription value in your installation that belongs to no remote user.

☞ For more information about this option, see "Using the Subscribe_by_remote option with many-to-many relationships" on page 140, and "Using the Subscribe_by_remote option with many-to-many relationships" on page 180.

- ◆ **Verify_threshold option** If the data type of a column is longer than the threshold, old values for the column are not verified when an UPDATE is replicated. The default setting is 1000.

This option keeps the size of SQL Remote messages down, but has the disadvantage that conflicting updates of long values are not detected.

- ◆ **Verify_all_columns option** The default setting is OFF. When set to ON, messages containing updates published by the local database are sent with all column values included, and a conflict in any column triggers a RESOLVE UPDATE trigger at the subscriber database.

The extraction utility for Adaptive Server Enterprise sets the public option in remote Adaptive Server Anywhere databases to match the setting in the Adaptive Server Enterprise database.

Examples

- ◆ The following statement sets the Verify_all_columns option to OFF in Adaptive Server Anywhere, for all users:

```
SET OPTION PUBLIC.Verify_all_columns = 'OFF'
```

- ◆ The following statements set the Verify_all_columns option to OFF in Adaptive Server Enterprise:

```
exec sp_remote_option Verify_all_columns, 'OFF'
go
```

In Adaptive Server Enterprise, replication options are used only by SQL Remote.

