

CHAPTER 13

Using SQL Remote with Replication Server

About this chapter

This chapter describes the additional components needed to use SQL Remote on an Adaptive Server Enterprise database that also participates in a Replication Server installation.

Contents

Topic	Page
When you need to use the SQL Remote Open Server	292
Architecture for Replication Server/SQL Remote installations	293
Setting up SQL Remote Open Server	296
Configuring Replication Server	299
Other issues	302

When you need to use the SQL Remote Open Server

The Message Agent for Adaptive Server Enterprise scans the Adaptive Server Enterprise transaction log to populate the stable queue, as described in the section "The stable queue" on page 279). SQL Remote messages are built from the transactions in the stable queue.

The Message Agent uses the same interface to scan the transaction log as the Replication Agent for Adaptive Server Enterprise. This means the Message Agent cannot scan the transaction log of an Enterprise database that is a primary site in a Replication Server setup (or a replicate site that allows asynchronous updates to primary data).

If there is a Replication Agent running against your Adaptive Server Enterprise database, you must use the SQL Remote Open Server as an additional component. In this case, SQL Remote is set up so that Replication Server populates the stable queue. The SQL Remote Message Agent does not scan the transaction log. Instead, the SQL Remote Open Server receives transactions from Replication Server. The transactions are parsed by the SQL Remote Open Server and stored in the SQL Remote stable queue.

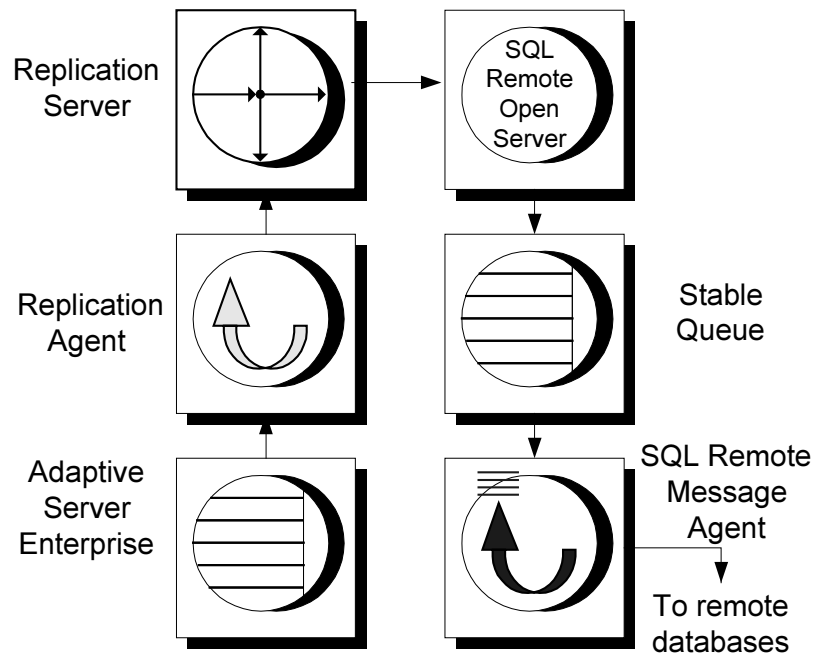
ℳ This chapter assumes knowledge of Replication Server. For information, see your Replication Server documentation.

Open Server runtime components required

The Open Server runtime components are not included with SQL Remote. You must obtain them separately from Sybase in order to use the SQL Remote Open Server.

Architecture for Replication Server/SQL Remote installations

The arrangement for using a database as a Replication Server primary site and as a SQL Remote database is illustrated in the following diagram. The diagram illustrates a case where the stable queue is held in a different database from the data being replicated. The stable queue may alternatively be held in the same database as the data being replicated. All connections are client/server connections, and so the components may be running on the same or different machines.

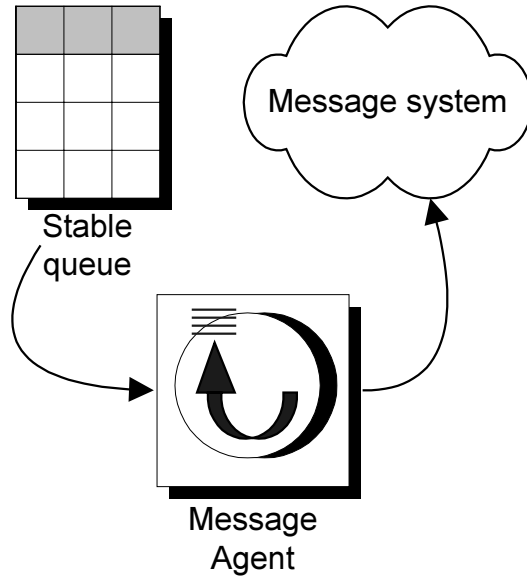


How the pieces fit together

The SQL Remote Open Server acts as a replicate database in the Replication Server setup, and so replication definitions and subscriptions are required in the Adaptive Server Enterprise database on all tables participating in SQL Remote replication and on several of the SQL Remote system tables.

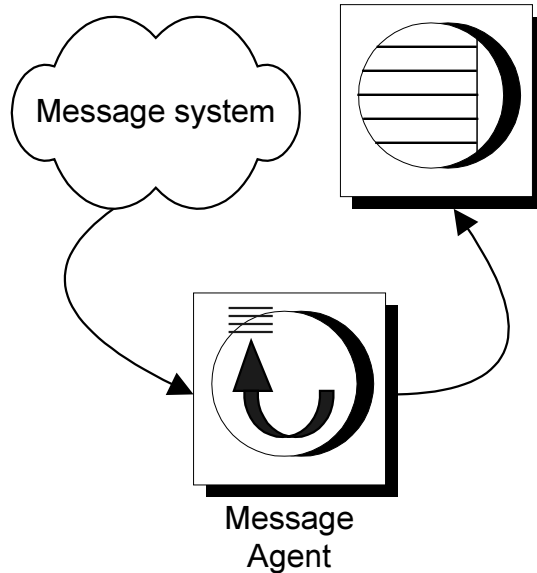
Contents of the stable queue

All operations are replicated to the SQL Remote Open Server, which stores them in the stable queue. The stable queue does not have copies of the tables being replicated. It parses the inserts, updates, and deletes to build transactions. All transactions are stored in an image column of a single table. These transactions are used by the Message Agent to build SQL Remote messages.



Incoming messages

The Message Agent always applies incoming SQL Remote messages directly to Adaptive Server Enterprise. It does not send operations to Replication Server. Incoming messages are applied directly to the consolidated database regardless of how the stable queue is populated. Conflict resolution is also performed in the same way.



Replication Server and SQL Remote

SQL Remote allows two-way replication between the consolidated database and remote databases. Replication Server is performing one-way replication from the consolidated database to the SQL Remote Open Server. From Replication Server's perspective, transactions that originate in remote SQL Remote databases appear as transactions originating in the consolidated SQL Remote database.

SQL Remote system tables

The SQL Remote Open Server requires information from the SQL Remote system tables concerning publications and subscriptions. The Open Server uses a connection to the Adaptive Server Enterprise database holding that information to retrieve it when it starts.

If the SQL Remote system tables are updated while the Open Server is running, the SQL Remote Open Server needs to receive this information at the correct time. For this reason, some of the SQL Remote system tables need to be marked for replication. This is described in "Setting up SQL Remote Open Server" on page 296.


The SQL Remote Open Server executable

The SQL Remote Open Server is the following executable:

- ◆ On Windows operating systems, the SQL Remote Open Server is *ssqueue.exe*.
- ◆ On UNIX operating systems, the SQL Remote Open Server is *ssqueue*.

Setting up SQL Remote Open Server

This section describes how to set up a SQL Remote installation using the SQL Remote Open Server. The procedure depends on whether the SQL Remote stable queue is being kept in a separate Adaptive Server Enterprise database from the tables being replicated or in the same Adaptive Server Enterprise database.

 For more information about stable queue location, see "The stable queue" on page 279.

Initial copies of the data

The setup procedure assumes you are using the extraction utility to produce an initial copy of the data in each remote database. You must be sure not to use the Replication Server materialization feature for this purpose.

The procedure for setting up SQL Remote Open Server has two stages:

- ◆ **Prepare a SQL Remote setup** This stage depends on whether you have an existing SQL Remote installation or not.
 - ◆ **Add the SQL Remote Open Server to the setup** This stage is the same regardless of previous installations.
- ❖ **To prepare your SQL Remote setup, if you have an existing SQL Remote installation:**
- 1 On a quiet primary database, use the Message Agent to scan any remaining transactions into the stable queue.

A quiet database is one where neither the Message Agent nor the SQL Remote Open Server is running, and where no transactions are being replicated.
 - 2 Follow the steps in the section "Upgrading SQL Remote for Adaptive Server Enterprise" on page 36 to upgrade your SQL Remote software at the consolidated site.
 - 3 Invalidate the Message Agent truncation point at the consolidated database using the following command:

```
dbcc settrunc('ltm', 'ignore')
```
 - 4 At the stable queue database, execute the stored procedure `sp_queue_log_transfer_reset`.
- ❖ **To prepare your SQL Remote setup, with no existing installation:**
- 1 Set up SQL Remote as described in "Setting Up SQL Remote" on page 29.

- 2 Set up your SQL Remote publications and subscriptions at this point. For information on this procedure, see "SQL Remote Design for Adaptive Server Enterprise" on page 159.
- 3 Extract the remote databases. For information on this procedure, see "Using the extraction utility" on page 209.

You are now ready to set up the SQL Remote Open Server.

❖ **To set up the SQL Remote Open Server:**

- 1 If the SQL Remote stable queue is in a separate database:
 - ◆ Set up the stable queue database as a replicate database in a Replication Server setup. This will create the tables and procedures needed by Replication Server, such as **rs_lastcommit**.
 - ◆ Drop the Replication Server connection to the stable queue database.
- 2 Add an entry to your interfaces file for the SQL Remote Open Server. The default name used on the SQL Remote Open Server command line is **SSQueue**.
- 3 Start the SQL Remote Open Server.
- 4 Create a Replication Server connection to the SQL Remote Open Server. The user ID and password for this connection must match the user ID and password specified on the SQL Remote Open Server command line for the stable queue connection (that is, the `-cq` switch, or `-c` if `-cq` is not specified).

Configure Replication Server now

You should configure Replication Server for this connection at this point. For a description, see "Configuring Replication Server" on page 299.

- 5 Define, activate, and validate Replication Server replication definitions and subscriptions for the SQL Remote tables **sr_marker**, **sr_remoteuser**, **sr_subscription**, and **sr_passthrough**. The script **ssremote.rs** is a sample script to perform this task. You will need to edit the server and database names in the script to match your names.

If the SQL Remote system tables have any data in them, create the replication definitions so that no materialization happens.

☞ For information on creating replication definitions with no materialization, see the *Replication Server Administration Guide*. The section in *Chapter 7, Managing Subscriptions* entitled Bulk Materialization describes how to set up Replication Server for the case where data exists at a remote database.

- 6 Define, activate, and validate replication definitions and subscriptions for the tables in your database that need to be replicated by SQL Remote. These must be created without materialization.

Configuring Replication Server

This section describes how to configure Replication Server for use with the SQL Remote Open Server

The Replication Server connection to the SQL Remote Open Server must have several configuration parameters set.

Set the `dsi_xact_group_size` parameter

By default, Replication Server groups multiple transactions into larger transactions. The `dsi_xact_group_size` parameter controls the maximum size of a grouped transaction.

The `dsi_xact_group_size` parameter must be set to `-1` to disable transaction grouping. Transactions that originate from different remote databases in a SQL Remote setup must not be grouped together.

How to set the parameter

You can set the parameter using the following statement:

```
CONFIGURE CONNECTION TO "ssqueue_server"  
SET dsi_xact_group_size TO '-1'
```

Set the `dsi_fadeout_time` parameter

For performance reasons, the SQL Remote Open Server does not commit changes to the stable queue after every transaction. The SQL Remote Open Server does commit changes to the stable queue when any of the following happens:

- ◆ One thousand transactions are sent to the SQL Remote Open Server
- ◆ Replication Server disconnects from the SQL Remote Open Server
- ◆ An operation on `sr_marker` replicates through to the SQL Remote Open Server

The commits to the stable queue are needed for the following reasons:

- ◆ Make newly replicated transactions available to the Message Agent sending process
- ◆ Release any page locks in the stable queue

The **dsi_fadeout_time** connection parameter controls the amount of idle time before Replication Server disconnects from a data server. This should be set low so the SQL Remote Open Server will commit changes to the stable queue whenever there is idle time. A value of 20 seconds is recommended.

How to set the parameter

You can set the parameter using the following statement:

```
CONFIGURE CONNECTION TO "ssqueue_server"  
SET dsi_fadeout_time TO '20'
```

Set the **dsi_num_threads** parameter

The SQL Remote Open Server does not support multiple DSI threads. Replication Server should not be configured to use multiple DSI threads on SQL Remote connections.

Create replication definitions for SQL Remote data

Replication definitions for tables being replicated by SQL Remote must have certain characteristics. This section describes those characteristics.

In some circumstances SQL Remote replicates an UPDATE operation as an INSERT or a DELETE (see "Replication of updates" on page 101). This is referred to as **subscription migration** in the Replication Server documentation. In order to replicate an UPDATE as an INSERT, SQL Remote requires the full pre-image of the row. This means that Replication Server must specify the values of every column in the WHERE clause of any UPDATE to a table that might need to be replicated as an INSERT.

The simplest way to achieve this is to list all columns in the PRIMARY KEY of the replication definition. This forces Replication Server to include every column in the WHERE clause of every UPDATE. REPLICATE MINIMAL COLUMNS can be used on these replication definitions to prevent every column from being listed in the SET clause of the UPDATE.

Using the dsi_data_style data style

Replication Server 11.5 has a new **dsi_data_style** for SQL Remote. This data style automatically includes all columns in the WHERE clause of every UPDATE. It is not necessary to list all columns in the PRIMARY KEY of the replication definition. A replication definition using REPLICATE MINIMAL COLUMNS prevents Replication Server from keeping the full pre-image of rows being updated, so the SQL Remote **dsi_data_style** will not work with REPLICATE MINIMAL COLUMNS.

Suspend and restart the connection

After configuring the Replication Server connection to the SQL Remote Open Server, you should suspend and resume the connection so that the parameter settings can take effect. The following commands accomplish this task:

```
suspend connection to ssqueue_name
go

resume connection to ssqueue_name
go
```

Other issues

This section lists other issues regarding using SQL Remote with Replication Server.

Running the Message Agent The Message Agent should be run with command line switches to receive and send (-r and -s). This will prevent the Message Agent from attempting to scan the transaction log. If the Message Agent attempts to scan the transaction log while the Replication Agent is running, it will get an error attempting to reserve the "log transfer context".

Procedure calls in SQL Remote Open Server The SQL Remote Open Server passes all procedure calls it receives from Replication Server through to the stable queue database. For example, **rs_get_lastcommit** and **rs_update_lastcommit** are executed in the stable queue database.

Coordinated dumps Replication Server provides a mechanism to coordinate database dumps and transaction log dumps between the main database and the stable queue database. The **rs_dumpdb** and **rs_dumptran** function strings can be used to perform coordinated dumps of the stable queue database. Please see the Replication Server documentation for more information.

Schema changes If you make any schema changes to a SQL Remote installation, you must do so on a quiet system. This includes shutting down the SQL Remote Open Server.