

CHAPTER 11

Replicating Data with SQL Remote

About this chapter This chapter shows you how to set up a simple replication system using SQL Remote.

Contents

Topic	Page
Tutorial overview	216
Setting up the consolidated database	217
Setting up the remote database	220
Start replicating data	222
Restoring the database and database settings	226

Tutorial overview

This tutorial describes how to set up a simple SQL Remote replication system.

Data replication is the sharing of data between physically distinct databases. By sharing information they become a dispersed database system.

With SQL Remote you can carry out two-way replication between a central database (called the **consolidated database**) and a set of **remote databases**. These remote databases may be on laptop computers, while the consolidated database may be on a network in an office. All the setup and administration is carried out at the consolidated database.

This tutorial describes a very simple case, replicating one table from the sample database to a single remote database on the same machine.

☞ For complete information about SQL Remote, see the book *Data Replication with SQL Remote*.

Tutorial goals

In the tutorial you act as the system administrator of a consolidated Adaptive Server Anywhere database. The replication system consists of a simple table from the sample database. The table we replicate is the **department** table, which is one of the simplest in the database.

The tutorial takes you through the following steps:

- 1 Setting up the consolidated database.
- 2 Creating a file-sharing replication system with a single remote database.
- 3 Replicating data between the two databases.

❖ To prepare for the tutorial:

- 1 Create a directory to hold the files you make during this tutorial; for example *c:\tutorial*.

```
mkdir c:\tutorial
```

- 2 Create a subdirectory for each of the two user IDs in the replication system, to hold their messages. Create these subdirectories using the following statements at a system command line:

```
mkdir c:\tutorial\dba
```

```
mkdir c:\tutorial\field
```

Setting up the consolidated database

This section of the tutorial describes how to prepare the consolidated database of a simple replication system.

Preparing a consolidated database for replication involves the following steps:

- 1 Create a message type to use for replication.
- 2 Grant PUBLISH permissions to a user ID to identify the source of outgoing messages.
- 3 Grant REMOTE permissions to all user IDs that are to receive messages.
- 4 Create a publication describing the data to be replicated.
- 5 Create subscriptions describing who is to receive the publication.

Notes

You require DBA authority to carry out these tasks.

Add a SQL Remote message type

All messages sent as part of replication use a message type. A message type description has two parts:

- ◆ A message link supported by SQL Remote. In this tutorial, we use the FILE link. This is a file sharing method, where the messages are left in a file on disk, and read by the intended recipient. SQL Remote also supports ftp and e-mail protocols.
- ◆ An address for this message link, to identify the source of outgoing messages. In this tutorial we use a file path to specify where the messages will be left.

Message types are created in all new databases, but you do need to supply an address for the message type you will use.

❖ To add an address to a message type:

- 1 From Sybase Central, connect to the sample database as user ID **dba** using the password **sql**, and open the sample database container (**asademo**) on the left panel.
- 2 Click the SQL Remote folder on the left panel.
- 3 Double-click the Message Types folder on the right panel.
- 4 Double-click the FILE message type.

- 5 Enter a publisher address to provide a return address for remote users. The publisher address is the directory you have created to hold messages for the consolidated database (*dba*).
- 6 Click OK to save the message type.

Add the publisher to the database

Each database in a SQL Remote replication system needs a single user ID that identifies the **publisher** of the data. Here, we make the **dba** user ID the publisher.

❖ To set the publisher:

- 1 Click the Users & Groups folder on the left panel.
- 2 Right-click the **dba** user, and select Set As Publisher from the popup menu. The **dba** Type column then displays the entry Publisher.

A database can have only one publisher. You can find out who the publisher is at any time by opening the SQL Remote folder.

Add a remote user to the database

Each remote database is identified in the consolidated database by a user ID with REMOTE permissions.

When a remote user is added to a database, the message system they use and their address under that message system need to be stored along with their database user ID.

❖ To add a remote user:

- 1 Click the SQL Remote folder in the **dba** group on the left panel, then click the Remote Users folder, also on the left panel.
- 2 Double-click Add Remote User on the right panel. The New Remote User wizard is displayed.
- 3 Create a remote user with user name **field**, password **field**, message type **file**, and address *field*. You should select the defaults for all other entries in the Wizard, except that you should grant DBA permissions as well as REMOTE DBA permissions to the user on the Indicate Authorities page. When you have finished all the entries, click Finish to create the remote user.

Notes

If you forget to set DBA permission in the Wizard, you can set it by double-clicking the user, and checking DBA in the Authorities tab.

You have now created the users who will use this system.

Add publications and subscriptions

This section describes how to add a publication to a database, and how to add a subscription to that publication for a user. The publication replicates all rows of the table **department**.

❖ To add a publication:

- 1 Click the Publications folder in the SQL Remote folder of the sample database.
- 2 Double-click Add Publication. The Publication Wizard is displayed.
- 3 Name the publication **DepartmentPub** on the first page of the Wizard.
- 4 On the next page, click Add Table and select **department** from the list. Leave All Columns selected, and press OK to add the table.
- 5 Complete the Wizard to create the publication.

Add a subscription

Each user ID that is to receive changes to a publication must have a **subscription** to that publication. Subscriptions can only be created for a valid remote user. You need to add a subscription to the **DepartmentPub** publication for the remote database user **field**.

❖ To add a subscription:

- 1 Open the Publications folder, which is in the SQL Remote folder.
- 2 Right-click DepartmentPub in the right panel, and select Subscribe For from the popup menu. A window is displayed, showing the **field** user in the list.
- 3 Select **field** and click Subscribe to subscribe the **field** user to the **DepartmentPub** publication.

You have now set up the consolidated database.

Setting up the remote database

The remote database needs to be created and configured in order to send and receive messages and participate in a SQL Remote setup.

The database extraction utility enables you to carry out all the steps needed to create a remote database complete with subscriptions and required user IDs.

You need to extract a database from the consolidated database for remote user **field**.

❖ To extract a database:

- 1 Click the Remote Users folder, in the SQL Remote folder.
- 2 Right-click the **field** remote user, and select Extract Database from the popup menu. The Extraction Wizard is displayed.
- 3 The first page informs you that you will extract from the running database **asademo**.
- 4 On the next page, choose to Start Subscriptions Automatically, for user **field**.
- 5 Create the database as file *c:\tutorial\field.db*, using a transaction log of name *field.log* in the same directory.
- 6 Choose to extract all parts of the schema (the default setting).
- 7 Leave the other options at their default settings, and create the remote database.

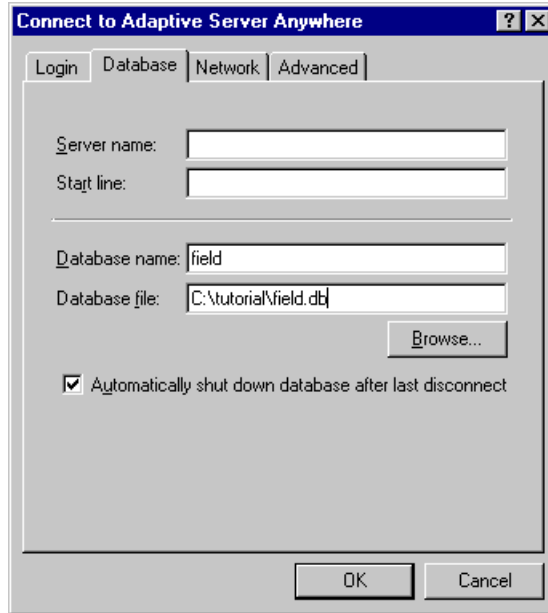
In a proper SQL Remote setup, the remote database **field** would be loaded on to the computer, together with a database server and any client applications required. For this tutorial, we leave the database where it is and use Interactive SQL to input and replicate data.

Confirm the database is created properly

To see what the extraction utility has done, connect to the *field* database and confirm that all the database objects are created.

❖ To browse through the remote database:

- 1 From Sybase Central click the connect button, and enter the user ID **field** using password **field** to connect to the database.



On the Database tab of the Connect to Database window, enter the database name **field** and the database path `c:\tutorial\field.db` to connect to the field database. Click OK to connect to the Database.

- 2 Open the tables folder. The **department** table, owned by user **dba**, is in the list.
- 3 Right-click the department table, and select View Data from the popup menu. Interactive SQL opens, showing that all the rows of the department table are filled (five rows). Close Interactive SQL.
- 4 Open the SQL Remote folder. The **field** user is displayed in the right panel, showing that **field** is the publisher for this database. Any data sent from this database comes from the user **field**, just as any data from the consolidated database comes from the user **dba**.
- 5 Open the Publications folder. You will see that the **DepartmentPub** publication is present.
- 6 Open the Remote Users folder. You will see that user ID **dba** is present, and is designated as a Consolidated user. This means **dba** is the publisher of the consolidated database (above the field database in the hierarchy).

You can browse around the rest of the database if you wish.

What next?

The system is now ready for replication.

Start replicating data

You now have a replication system in place. In this section, data is replicated from the consolidated database to the remote database, and from the remote to the consolidated database.

Enter data at the consolidated database

First, enter a row at the consolidated database.

❖ **To enter data at the consolidated database:**

- 1 Connect to the sample database from Interactive SQL as user ID **dba**, with password **sql**.

- 2 Enter and commit a row into the **department** table:

```
INSERT INTO department (dept_id, dept_name)
VALUES (202, 'Eastern Sales') ;
COMMIT ;
```

- 3 Confirm that the data is entered by viewing the data in the tables:

```
SELECT *
FROM Department
```

The next step is to send the relevant rows to the remote database.

Send data from the consolidated database

To send the rows to the remote database, you must run the Message Agent at the consolidated database. The *dbremote* program is the Message Agent for Adaptive Server Anywhere.

❖ **To send the data to the remote database:**

- 1 From a command prompt, change to your tutorial directory. For example,

```
> c:
> cd \tutorial
```

- 2 Enter the following statement at the command line to run the Message Agent against the consolidated database:

```
dbremote -c "dbn=asademo;uid=dba;pwd=sql"
```


Use the absolute path (above) to link to the database file or use the path which is relative to your server. For example, if your server is in the *C:\tutorial\dba* directory, you can use the statement:

```
dbremote -c "dbn=..\asademo.db;uid=dba;pwd=sql"
```

- 3 Click Shutdown on the Message Agent window to stop the Message Agent when the messages have been sent. The Message Agent window displays the message Execution completed when all processing is complete.

Receive data at the remote database

To receive the insert statement at the remote database, you must run the Message Agent, *dbremote*, at the remote database.

❖ To receive data at the remote database:

- 1 From a command prompt, change to your tutorial directory. For example,

```
> c:
> cd \tutorial
```

- 2 Enter the following statement at the command line to run the Message Agent against the **field** database:

```
dbremote -c
"dbf=C:\tutorial\field.db;uid=field;pwd=field"
```

Use the absolute path (above) to link to the database file or use the path which is relative to your server. For example, if your server is in the *C:\tutorial\dba* directory, you can use the statement:

```
dbremote -c "dbf=..\field.db;uid=field;pwd=field"
```

- 3 Click Shutdown on the Message Agent window to stop the Message Agent when the messages have been processed. The Message Agent window displays the message Execution completed when all processing is complete.

Notes

The Message Agent window displays status information while running. This information can be output to a log file for record keeping in a real setup. You will see that the Message Agent first receives a message from *asademo*, and then sends a message. This return message contains confirmation of successful receipt of the replication update; such confirmations are part of the SQL Remote message tracking system that ensures message delivery even in the event of message system errors.

Depending on the current status of the database, there are three different connectivity parameters you can use to connect to a database:

- ◆ **dbf** Connect to the database using the database file. This parameter requires you to specify the database file itself. If no server is currently running, a server will be started and the database will be loaded onto it. If a server is already running, the database will be loaded onto the default server.
- ◆ **dbn** Connect to the database using the database name. This parameter requires you to specify the name of the database. You can only use this parameter when the database is already running.
- ◆ **dbsource** Connect to the database using the database source. A data source is a collection of parameters stored in the system registry or in a set of files. The source is referenced simply by its name.

☞ For more information about data sources, see "DataSourceName connection parameter" on page 46 of the book *Adaptive Server Anywhere Reference Manual*.

Verify that the data has arrived

Connect to the *field* database using Interactive SQL, and inspect the **department** table, to see that the row has been received.

❖ To verify that the data has arrived:

- 1 Connect to the *field* database using Interactive SQL.
- 2 Inspect the **department** table by typing the following statement:

```
SELECT *  
FROM dba.department
```

You will see that the **department** table contains the row entered at the consolidated database.

Replicate from the remote database to the consolidated database

Now try entering data at the remote database and sending it to the consolidated database. Only the outlines are presented here.

❖ To replicate data from the remote database to the consolidated database:

- 1 Connect to the **field** database from Interactive SQL.

- 2 INSERT a row at the remote database. For example

```
INSERT
INTO dba.Department (dept_id, dept_name)
VALUES (203, 'Western Sales')
```

- 3 COMMIT the row.

```
COMMIT;
```

- 4 With the *field.db* database running, run *dbremote* to send the message to the consolidated database.

```
dbremote -c "dbn=field;uid=field;pwd=field"
```

- 5 With the *asademo.db* database running, run *dbremote* to receive the message at the consolidated database:

```
dbremote -c "dbn=asademo;uid=dba;pwd=sql"
```

- 6 Connect to the consolidated database and display the **department** table:

```
SELECT *
FROM department
```

Restoring the database and database settings

Once you have completed the tutorial it is important to undo any changes you have made to the database. Make sure that the following steps are completed in order to insure that the settings are reset properly.

❖ **To delete the remote user:**

- 1 Click the Remote Users folder for the sample database (dba) on the left panel.
- 2 Right-click the field user, and select Revoke Remote from the popup menu. Click Yes to remove the user from the list.

❖ **To delete the publication:**

- 1 Click the Publications folder in the SQL Remote folder.
- 2 Right-click **DepartmentPub** in the right panel and select Delete from the popup menu. Click Yes to confirm the deletion.

❖ **To revoke the publishing status from the sample database:**

- 1 Click the Users and Groups folder on the left panel.
- 2 Right-click the DBA user on the right panel, and select Revoke Publisher from the popup menu.

Revoking a status while running the database

The status of a user cannot be modified while the user is running the database. Make sure that the user is idle before revoking or invoking a status.

❖ **To restore the original message type settings:**

- 1 Click the Message Types folder in the SQL Remote folder.
- 2 Right-click the File message type in the right panel and select Properties.
- 3 Delete the publisher address and click OK to restore the settings.

❖ **To delete the remote database:**

- 1 Click the Utilities folder on the left panel.
- 2 Double-click Erase Database from the right panel. The Erase An Adaptive Server Anywhere Wizard is displayed.

- 3 Enter the name of the database you want to erase or click Browse to search for the database. If you followed the tutorial correctly, the database should be stored in your *tutorial* directory and called *field.db*.
- 4 On the next page, choose to delete a database file and click next to finish erasing the file.

Notes

- ◆ Now that the tutorial is complete, it might be a good idea to delete the tutorial directory (*C:\tutorial*) in order to save space.

Restore the data in the database:

The most important part of the cleanup process is to ensure that the changes to the sample database are reversed. The integrity of the sample database is very important in order to carry out other tutorials in later chapters of the manual.

❖ To delete the inserted data from the sample database:

- 1 Connect to the sample database (dba) using Interactive SQL.
- 2 Type the following statements in the command window and click Execute:

```
DELETE
FROM department
WHERE dept_id = 202
   OR dept_id = 203
   OR dept_name = 'Eastern Sales'
   OR dept_name = 'Western Sales';
COMMIT;
```

- 3 Confirm that the data has been deleted by viewing the data in the tables:

```
SELECT *
FROM department
```

The inserted data has been deleted and the database has been restored to its original state.

Notes

The previous statements delete only the data that was added specifically in the tutorial. If you have modified the database in any other way, you must reverse the changes you have made.

