Replication setup for DB2 Universal Database

A step-by-step approach to "user-copy" replication

Mika Nikolopoulou

May 27, 2004

Replication is one way to make your data available where and when you need it. This article explains what replication can do for you, and shows you how to set it up, step by step, using an example from an actual customer site. The author includes hints and tips along the way to help ensure your success.

Introduction

Data replication is an important technique for ensuring data availability in the on demand business environment. It's an asynchronous, log-based process that permits you to copy data from one location to another, keeping the data in the second location identical to the first. There are many reasons why you might want to replicate data, but very often it's used for query load balancing, for disaster recovery, to reduce latency and bring the data closer to the user, or to consolidate data from multiple sources. Because it is based on the database logs it is very efficient in terms of performance compared with other methods of moving data.

In DB2® Universal Database™ (UDB), SQL replication capability is included in the base product. From the Control Center you can access the Replication Center, a graphical interface for the setup of replication. There is also a scripting language for replication called ansclp which allows you to create scripts to automate replication setup.

This article describes a specific example of how to set up data replication using the Replication Center. In addition, you'll see how to set up and use ansclp. I've based the information in this article on a real replication implementation at a customer site.

Please note that the case I describe is a specific implementation of replication between a UNIX® server and a Windows® server, and was designed to meet one customer's needs. This configuration and setup does not apply to all situations, but the concepts and the steps are the same for most cases. You can adapt this scenario using your own database names, server names, aliases, paths, and so on.

Architecture for our replication example

In this implementation we are using two servers:
Server 1 is a Sun server with Solaris 5.2 operating system. The server's host name is DAISY, and it has 4 processors and 2 GB of memory. We have installed DB2 V8.1 and fix pack 2. This is an online transaction processing (OLTP) system with about 500 BM of data. This data is mainly information about trouble tickets from the customer technicians who are traveling to various cities to fix problems for the company's products. This is a heavily used system with more than 60,000 updates, inserts or deletes per day. Each update is about 1KB, and the majority of the updates will take place between 10:00 PM and 8:00 AM daily. So within a window of 10 hours per day, there will be approximately 60 MB of activity, at a rate from about 6 MB per hour to a maximum of 8 MB per hour. This data will be replicated to a second server. This replication volume can be handled with acceptable performance in a replication environment. The communication between the two servers that are going to participate in the replication is a 10BaseT connection.

Server 2 is a Windows 2000 server, with 2 CPUs, 1GHz each, 1 GB of memory and 40 GB of disk. It is loaded with DB2 V8.1 and fix pack 2. The host name for this server is KEEL. Both servers are running with one partition only. This data on this server will be a read-only replica of the main Sun server, and will be accessed by reporting tool users for reporting only. Because this server is a read-only server, the replication does not need to take care of any update conflict scenarios and does not need to maintain referential integrity (RI).

The Sun server will be the capture server, and the Windows server will be the apply server. You will see how to set up the replication environment using the Replication Center of DB2, which is the graphical user interface (GUI) for setting up replication. In this example, we will run the Replication Center from the Windows server, which will be the control server for replication as well as being the apply server.

User copy replication is one of the options you have for setting up replication. User copy is the default type of the target table. It is a read-only target table that includes only those columns defined in the subscription-set member (which is the definition of the replication set). A user-copy table can have the same structure as the source table or it can have a subset of source columns, with or without before images or calculated columns.

Figure 1 shows the architecture for this replication scenario.

**Figure 1. Architecture of the replication scenario**

**Unix server**
DAISY

**Windows server**
KEEL

---

**Source Replication Server**

**Source Database:** PPMT_DB
with local alias: DAISY_DB

**Target Replication Server**

**Target Database:** KEEL_DB

Catalog local PPMT_DB as DAISY_DB
Catalog remote KEEP_DB as KEEL_DB

Create database KEEP_DB for target database

---
Set up for replication

In this section we'll look at how to set up for replication, step by step.

Step 1 - Catalog a new alias name on the source database

In our case, on the server DAISY, we have a database called PPMT_DB. In order to properly use this database with the alias name DAISY_DB on the remote Windows server, we need to catalog an alias name DAISY_DB on the DAISY server, so that the capture process will not conflict with the name PPMT_DB.

On the Sun server, use the command:

```
catalog database PPMT_DB as DAISY_DB on /ppmt/db2ppmt
```

Note, the syntax of the command is: `catalog database dbname as aliasname on path`. If you need more details for this command, please see the DB2 UDB Command Reference.

If you fail to add this alias to the capture server, replication may not work properly. Adding this alias will ensure that the alias DAISY_DB is known to both the capture and the apply servers. Use the `catalog local node` command in order to catalog the local server and then the `catalog database` command to give an alias to PPMT_DB.

Please note that adding this alias is not necessary if you do not want to use a different name for the source database on the target server. In other words, if you do not mind giving the alias PPMT_DB on the target server for the source PPMT_DB database, then you do not need to create a local alias on the source server. Using the same name for the source database on both servers is actually recommended.

Step 2 - Catalog the source server from the DB2 Control Center

1. Using the DB2 Control Center on the Windows system, register the remote (source) database, so that it can be accessed locally by the Windows server. In the Control Center right click on All Cataloged Systems and add a new system; in our case it will be DAISY.
2. As soon as you see the new DAISY system in the Control Center, then right click on Instances and choose Add, in order to add a new database. In the window that comes up, enter the appropriate information for the source database system, including port number – the default is 50000 – instance name, and so on. With the Discover option, DB2 should give you the list of databases that are available on the remote server. Choose the one you want to access from your local environment and add it to the system. Now you have cataloged a system (node) and a database. In the Control Center, you should see now the system DAISY, and under Instances you should see the node DAISYNOD (we always give a name to the “node” that is the same as the remote system, and in this case we gave the name DAISYNOD, while the underlying instance name is db2ppmt). Then under DAISYNOD, click on Databases and you should have DAISY_DB as the local database name/alias for the remote database DAISY_DB.
Step 3 - Register a source server

1. From the Replication Center, expand the Replication Definitions, and right click on Capture Control Servers, then choose Add. In the Add Capture Control Servers window, add the server that you want to be used as a source. The term capture server in the Replication environment means the capture database. Check the checkbox next to DAISY_DB in order to use the remote Sun server database PPMT_DB with alias DAISY_DB, as your capture database.

Figure 2. Register the source database

2. Click OK. Now your capture control servers should have a new server called DAISY_DB.

Step 4 - Configure the profile for the capture control tables

Before creating the Capture Control Tables, we want to configure the environment so that we design it the way we want. In our environment we want all capture control tables to be stored in a single table space and not on separate table spaces as is the default. Also we want to change the schema name for the capture control tables.

1. Right click on the DAISY_DB Replication source name, and choose Manage Source Object Profiles.
2. In the window that opens, under the tab CD Tables, choose Specific schema and not Use the same schema as the source tables. Enter the schema name ASN1. Make sure that Allow full refresh for target table is checked.
3. Go to the tab CD-Table Spaces. Choose Put all CD tables in the same table space. Give a name for the tablespace. In our case this will be TSCD.
4. Also choose the default bufferpool, IBMEFAULTBP, and choose the page size to be 4. Depending on your database you may want to have a different page size that matches the page size of your base source tables. In this case we choose the default.

5. Go to the **CD-Table Indexes** tab. The CD Indexes can have the same schema and tablespace as the CD tables. In the Index Name, choose “Source table name”. This will help with name conflicts for CD indexes, because there is a limitation that will be fixed. Use caution here because the indexes have a limit of 18 characters, and the DB2 replication will truncate any CD index to 18 characters. Note, this is a current limitation that has been reported and will be fixed, because normally this is not a valid limitation for DB2 indexes. The normal limit is 128 characters.

6. In the **Truncation Rules** tab, leave the default values so that the Capture program truncates the index name from the right in case the name is too long.

In general, when you generate the SQL from the Replication Center, which will create the CD tables and indexes, be careful with name conflicts with indexes. You should check each index name to see if it conflicts with the same name on any index before running the generated SQL. The Replication Center will not do this check before running the SQL, and the SQL script will then run into the conflict and return an error message. In order to avoid this problem, you should do the check before running the SQL script for the registration (CD) tables and indexes creation.

**Figure 3, Figure 4, and Figure 5** show the screens for configuring the source database.

**Figure 3. Configure source database staged tables**
Step 5 - Create replication control tables for the capture program

1. Right click on Capture Control Servers and select Create Capture Control Tables, then Custom. Choose the database name, in our case DAISY_DB, and click OK.
2. In the next window, Create Capture Control Tables, you define the tablespace and tables for the Capture control tables. You should see by default the schema name ASN1 since this is the schema name we defined in our profile. If it is not there, then enter ASN1 for the schema.

3. For the table spaces TSASN1CA and TSASN1UOW, change the size to 512 MB instead of the default which is 10 MB. Choose the IBMDEFAULTBP for the bufferpool.

4. Click **OK**. A message window appears, containing information about the creation of the tables. Scroll to the end to see if there are any errors.

5. If there are no errors, then click **OK**. You should see a script of SQL commands that will generate the tables for the capture control server.

6. You have the option to run it now as it is, or save it in a script or file and run it later. Click on **Apply** or **OK** if you choose to run it now. The creation of the Registration control tables should now be completed.

**Step 6 - Enable replication**

1. To enable the database for replication, first ensure that the LOGRETAIN parameter in the database configuration for the capture server is set to RETAIN. You can check this parameter by issuing the following commands from the DB2 command line processor on the Sun server (DAISY):

   ```
db2 connect to DAISY_DB
db2 get db cfg
```

2. If LOGRETAIN is not already set to RETAIN, you may set it with the following command:

   ```
db2 update db cfg using LOGRETAIN RETAIN
```

3. Next you need to back up the database using the BACKUP command. If you have changed the LOGRETAIN value to RETAIN, the database will be in backup pending state. RETAIN means that the logging is no longer circular but archival, and therefore you need to have an initial backup before you continue.

   You may need to stop and start the database server in order for this configuration change to take effect. Issue the command `db2 force applications all` and then `db2stop`. Then `db2start` and the change should take effect.

4. Next expand the Capture Control Servers folder on the Replication Center, right click on the DAISY_DB database, and select **Enable Database for Replication**. Click **OK** on the window that comes up. In case you need to back up the database, you can do so either from the Replication Center by clicking **Backup now**, or through the Control Center.

**Step 7 - Register the tables**

The next step is to register the tables on the capture server that you want to replicate. In our case, we want to replicate 141 tables, and we will not select any columns or use any WHERE clauses to restrict our replication set. For each table you want to register, there will be a change data (CD) table and CD index created. As we defined in the profile, all these tables will be stored in the same tablespace. The tablespace should have enough space for all 141 CD tables. These tables will contain all changes that are made to the source tables, so be careful to allow for enough space to hold the changes.

1. Right click on **Registered Tables** and choose **Register Tables**.

2. In the Add Register Tables window, using your search criteria, search for the tables you want to register for replication. In our case, because we are using all tables with the PPMT_DB
database, we put the PPMT value next to the SCHEMA column, and the comparison should be "/=".

3. Then click Retrieve and you should get a list of all tables with PPMT schema name.
4. Choose all of them by pressing the Shift key and scrolling the mouse from the top of the list to the end. Then release the Shift. All tables should be selected and highlighted. Then click OK.
5. The next window comes up with the list of tables. Keep the defaults in this window and click OK. In the message pop up window, check to see if there are any error messages, and if not, click OK.

6. Now check the script. This script contains the SQL for the creation of all CD tables and indexes for the tables you are registering. It also updates the capture control tables.

7. Before running the script, check to see if there are any conflicts with the index names. Make sure that in the script no two index names are the same; otherwise the script will return errors. You can do this by first saving the script to a file using the Save to a File radio option and clicking OK. If you make any changes, transfer them to the script and run it from the task center, from the command center, or from the same replication window.

8. After you have run the script, you should be able to see all registered tables under the Registered Tables path in the DAISY_DB under the Capture Control Servers in the Replication Center. You can check in the Control Center by going to the DAISY_DB database and checking for the new CD tables.

**Step 8 - Create replication control tables for the Apply program**

1. Right Click on Apply Control Servers and select Create Apply Control Tables, Custom.
2. Choose the target database, in this case the KEEL_DB. Click OK.
3. In the window that comes up for Create Apply Control Tables, keep the default and press OK. The apply tables do not need a lot of space, so the 10MB space should be sufficient. You can add more space if you need to.
4. Close the message window if there are no errors, and then use the Run Now option to run the script. Click OK. After the script is run successfully, expand the Apply Control Servers folder. The KEEL_DB database should be displayed under the folder. The script updates the capture control tables and creates the apply control tables.

**Step 9 - Create the target database**

Create a new database on the target system, KEEL in our case. Name this database KEEL_DB. Use the default settings. You can use the DB2 command:

```
create database KEEL_DB
```

Or, use the Create Database Wizard from the Control Center and use all default settings.

**Step 10 - Create a subscription set**

Expand the Apply Control Servers folder, expand the KEEL_DB folder and right click the Subscription Sets folder. Select Create. In the Set Information tab:

1. Choose KEEL_DB as the apply control server alias.
2. Use a set name of your preference, in our case PPMT_TARGET.
3. Use a qualifier of your preference, in our case QUAL.
4. Choose the capture control server alias, in our case DAISY_DB.
5. The capture schema here is ASN1. Check the **Activate the subscription set** check box and select the **Make active indefinitely** radio button, as shown in Figure 6.

**Figure 6. Configure staged tables for source database**

6. Go the **Source-to-Target Mapping** tab and click **Add** to add members to the subscription set.
7. On the next window click on **Retrieve All**, to retrieve all the tables in the database, or make a selection based on criteria that you define at the top of the window.
8. Choose the tables you want to add in the subscription set. In our case we added three tables.
9. Right click on each table and click on **Change**. In this window you will find the mapping of columns, and you will have the option to restrict the data that will be replicated in case this is a requirement, using a WHERE clause or row filters.
10. You can also define the table space and the target Load options. In our case, keep the defaults and then click **OK**.
11. Click **OK** to create the subscription set, close the message window and run the script.

If you expand the apply control servers, you will be able to see the new Subscription set and its members.

**Step 11 - Create the password files for capture and apply**

For each apply server we need to have a password file. For each database, we need to have an entry in the password file with the id and password for accessing that database. In our case we have only one apply server – KEEL_DB –, and we have two databases, DAISY_DB and KEEL_DB; therefore we must have one password file with two entries on the KEEL_DB server.
1. On the Windows server, create the directory repl under SQLLIB, then from a command window enter the following command:

```
asnpwd init using "c:\Program Files\SQLLIB\repl\asnpwd.aut"
```

2. Then populate the password file with entries for the two databases using the following commands:

```
asnpwd add alias DAISY_DB id db2ppmt password db2ppmt using "c:\Program Files\SQLLIB\repl\asnpwd.aut"
asnpwd add alias KEEL_DB id db2admin password db4admin using "c:\Program Files\SQLLIB\repl\asnpwd.aut"
```

Start the Capture and Apply programs

It is preferable to start the capture and apply programs from the command line during testing. You can do this with simple commands, and you will have the opportunity to check immediately to see whether capture and apply started successfully. Once testing is complete, you should set up capture and apply as Windows servers using the asnscrt command documented in the DB2 V8 Replication Guide and Reference.

1. For the first time you want to start the capture, on the DAISY server, from the UNIX command prompt, type the following:

```
asncap CAPTURE_SERVER=DAISY_DB CAPTURE_SCHEMA=ASN1 CAPTURE_PATH=/ppmt/db2ppmt/db2ppmt startmode=COLD
```

2. The first time you start replication and when the database is empty, you want to start with the COLD option. The reason is that this way a full refresh will be made, and the empty tables will be populated automatically by the apply process using INSERTS instead of using the replication tables. This method is a preferred method for fast population of the tables. For the very large tables, you should use the ASNLOAD option.

3. At all other times, use the following command:

```
asncap CAPTURE_SERVER=DAISY_DB CAPTURE_SCHEMA=ASN1 CAPTURE_PATH=/ppmt/db2ppmt/db2ppmt/ startmode=WARMSI
```

Then a message appears and the server should be started.

4. To start the apply process, go to a command window in Windows and issue:

```
asnapply CONTROL_SERVER=KEEL_DB APPLY_QUAL=QUAL APPLY_PATH="c:\Program Files\SQLLIB\repl"
```

You need to specify the APPLY_PATH because you have to run the apply program from the location of the password file, or specify the location of the password file, otherwise connection to the database will fail.

5. In case you need to troubleshoot, a good method is to issue the Apply with the following options:

```
asnapply CONTROL_SERVER=KEEL_DB APPLY_QUAL=QUAL APPLY_PATH="c:\Program Files\SQLLIB\repl" trcflow copyonce > filename.out
```

This way, a trace file will be generated and with the copyonce option the apply will only take one cycle and will stop. Check the filename.out file for any messages to troubleshoot problems.
Stop the capture and apply

- To stop capture, issue the following command from the Sun server:
  
  ```bash
  asnccmd CAPTURE_SERVER=DAISY_DB CAPTURE_SCHEMA=ASN1 STOP
  ```

- To stop apply, issue the following command on the Windows server:
  
  ```bash
  asnacmd CONTROL_SERVER=KEEL_DB APPLY_QUAL=QUAL STOP
  ```

Check the replication

1. To verify that replication is working, add a couple of new rows to one of the source tables using the INSERT command from the Sun server command prompt:

   ```sql
   DB2 INSERT INTO tablename VALUES ('xxx', 'yyy')
   ```

2. Check in the source table that the new rows are properly inserted. Then wait for the next replication cycle, and check the target table. The target table should be updated with the new rows.

How to setup replication using asnclp

Step 1 - Set up the asnclp command line

In order to do this in Windows, it is important for the system environment path to have the right parameters for Java.

Go to Start, Programs, Settings, Control Panel, then System. In System go to the Advanced tab, and choose Environment Variables.

The CLASSPATH variable should look like this:

```
.;C:\PROGRA~1\IBM\SQLLIB\java\db2java.zip;C:\PROGRA~1\IBM\SQLLIB\java\db2jcc.jar;
C:\PROGRA~1\IBM\SQLLIB\java\sqlj.zip;C:\PROGRA~1\IBM\SQLLIB\bin;c:\\PROGRA~1\IBM\SQLLIB\java\Common.jar;c:\\PROGRA~1\IBM\SQLLIB\tools\db2replapis.jar;c:\\PROGRA~1\IBM\SQLLIB\tools\db2cmdn.jar;c:\\PROGRA~1\IBM\SQLLIB\tools\jt400.jar;c:\\PROGRA~1\IBM\SQLLIB\java\;C:\PROGRA~1\IBM\SQLLIB\java\jdk\bin;
```

The PATH variable should look like this:

```
%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;
C:\Program Files\IBM\Trace Facility;C:\Program Files\Personal Communications;
C:\Notes;C:\Utilities;c:\\PROGRA~1\IBM\SQLLIB\BIN;c:\\PROGRA~1\IBM\SQLLIB\FUNCTION;c:\\PROGRA~1\IBM\SQLLIB\SAMPLES\REPL;c:\\PROGRA~1\IBM\SQLLIB\java\;C:\PROGRA~1\IBM\SQLLIB\java\jdk\bin;
```

If you make any changes, reboot the machine for these setting changes to take effect.

Step 2 - Start asnclp

Open a DB2 Command Line Processor window from the DB2 menu on your Windows server (Start, Programs, IBM DB2, Command Line Tools, Command Window) and then type asnclp and enter. You should get the prompt : Repl>
Now you can start typing the asnclp commands. The following commands will add members to an existing subscription set. This will expedite the subscription set creation process, especially if you have many members and many servers.

```
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET CAPTURE SCHEMA SOURCE "ASN1";
*** Comment: (this is required because we have chosen to use a different schema than ***
*** the default schema which is ASN)***
SET SERVER CAPTURE TO DB DAISY_DB DBNAME DAISY_DB ID
db2ppmt PASSWORD db2ppmt;
SET SERVER CONTROL TO DB KEEL_DB ID  db2admin PASSWORD "db4admin";
SET SERVER TARGET TO DB KEEL_DB ID  db2admin PASSWORD "db4admin";
SET OUTPUT CAPTURE SCRIPT "C:\SQLLIB\repl\CAP.LOG";
SET OUTPUT CONTROL SCRIPT "C:\SQLLIB\repl\CNTL.LOG";
SET OUTPUT TARGET SCRIPT "C:\SQLLIB\repl\TARGET.LOG";
CREATE MEMBER IN SETNAME PPMT_TARGET APPLYQUAL QUAL SOURCE
PPMT.BAD_PAIR_FIXED TARGET NAME PPMT.TGBAD_PAIR_FIXED DEFINITION
IN TGTSPPM TC USERCOPY COLS ALL REGISTERED;
```

You can repeat the last command, CREATE MEMBER, for as many times as the number of members you need to add in each subscription set.

**Conclusion**

I've showed you the basic steps for setting up replication. Plug in your own server and database names to try replication in your own environment, and soon you will be taking advantage of the replication capabilities in DB2 Universal Database to provide on demand availability of your data, wherever and whenever it is required.

**Acknowledgement**

I would like to thank replication experts Carol Rigdon and John Casey for their invaluable help and support during the writing of this article.
Related topics

- For more assistance with replication, refer to the DB2 V8 Replication Guide and Reference.
- The redbook A Practical Guide to DB2 UDB Data Replication V8 provides detailed information on setting up replication in many different environments.
- For help with troubleshooting, refer to DB2 UDB Data Replication Troubleshooting using DB2 Administration Client and Other Methods.

© Copyright IBM Corporation 2004
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)