IBM Information Integration
Version 9.5

ASNCLP Program Reference for Replication and Event Publishing
Note

Before using this information and the product that it supports, read the information in "Notices" on page 197.
# Contents

**Chapter 1. Getting started with the ASNCLP program** ........................................... 1
- Supported operating systems .................................................................................. 1
- Setting up a Java environment to run the ASNCLP program .................................. 1
- Binding z/OS packages for the ASNCLP program .................................................. 2
- Running the ASNCLP commands in interactive mode .......................................... 2
- Running the ASNCLP commands in execute-immediately mode ........................... 3
- Running the ASNCLP commands using an input file ............................................. 3
- ASNCLP configuration file ...................................................................................... 4

**Chapter 2. ASNCLP commands for SQL replication** .............................................. 7
- Sample ASNCLP scripts for setting up SQL replication ........................................ 8
- ALTER MEMBER ADD COLS command .............................................................. 11
- ALTER REGISTRATION command .................................................................. 12
- ALTER SUBSCRIPTION SET command ........................................................... 15
- ASNCLP SESSION SET TO command (SQL replication) ................................... 17
- CREATE CONTROL TABLES FOR command (SQL replication) ......................... 18
- CREATE MEMBER command ............................................................................. 20
- CREATE REGISTRATION command ................................................................. 28
- CREATE STMT command .................................................................................... 32
- CREATE SUBSCRIPTION SET command ........................................................ 34
- DROP CONTROL TABLES ON command ......................................................... 36
- DROP MEMBER command ............................................................................... 37
- DROP REGISTRATION command .................................................................... 38
- DROP STMT command ..................................................................................... 38
- DROP SUBSCRIPTION SET command ............................................................. 39
- OFFLINE LOAD command ................................................................................. 40
- PROMOTE REGISTRATION command ............................................................ 40
- PROMOTE SUBSCRIPTION SET command ....................................................... 42
- SET CAPTURE SCHEMA command (SQL replication) ......................................... 44
- SET DROP command (SQL replication) ............................................................. 45
- SET LOG command .......................................................................................... 46
- SET OUTPUT command (SQL replication) ....................................................... 47
- SET PROFILE command (SQL replication) ...................................................... 47
- SET RUN SCRIPT command (SQL replication) ............................................... 51
- SET SERVER command (SQL replication) ......................................................... 54
- SET TRACE command ...................................................................................... 56

**Chapter 3. ASNCLP commands for Q replication and Classic replication** .......... 57
- Sample ASNCLP scripts for Q replication .......................................................... 57
  - Sample ASNCLP scripts for setting up unidirectional Q replication ................ 57
  - Sample ASNCLP scripts for setting up unidirectional Q replication from a Classic data source .......................................................... 61
- Sample ASNCLP scripts for setting up bidirectional Q replication ..................... 63
- Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers) ... 68
- Sample ASNCLP scripts for setting up peer-to-peer Q replication (three servers) ... 73
- Sample ASNCLP script for promoting unidirectional configurations ............... 79
- Sample ASNCLP scripts for promoting peer-to-peer configurations ................ 80
- ASNCLP commands for unidirectional Q replication ........................................ 82
  - ALTER CAPTURE PARAMETERS command (Classic replication) ................ 84
  - ALTER CONFIGURATION APPLY command ............................................. 84
  - ALTER QSUB command (unidirectional replication) .................................... 85
  - CREATE QSUB command (unidirectional replication) .................................... 89
  - DROP QSUB command (unidirectional Q replication) ................................... 102
  - LIST QSUB command (Q replication) ......................................................... 103
  - LIST REPLQMAP command (Q replication) ............................................... 105
  - SET DROP command (unidirectional replication) ........................................ 106
- ASNCLP commands for multidirectional Q replication ........................................ 107
  - ALTER QSUB command (bidirectional replication) ...................................... 109
  - ALTER QSUB command (peer-to-peer replication) ....................................... 111
  - CREATE QSUB command (bidirectional replication) .................................... 113
  - CREATE QSUB command (peer-to-peer replication) .................................... 116
  - DROP SUBGROUP command (multidirectional Q replication) ...................... 118
  - DROP SUBTYPE command (bidirectional replication) ................................... 118
  - DROP SUBTYPE command (peer-to-peer replication) ................................... 119
  - LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication) ... 120
  - SET BIDI NODE command ............................................................................ 121
  - SET CONNECTION command (multidirectional Q replication) .................... 122
  - SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication) .......................................................... 123
  - SET MULTIDIR SCHEMA command (multidirectional Q replication) .......... 124
  - SET OUTPUT command (multidirectional Q replication) .............................. 125
  - SET PEER NODE command ......................................................................... 125
  - SET REFERENCE TABLE command (multidirectional Q replication) .......... 127
  - SET SERVER command (multidirectional Q replication) ............................... 128
Chapter 1. Getting started with the ASNCLP program

The ASNCLP program is a command-line interface for administration of SQL replication, Q replication, Classic replication, and relational event publishing. Some commands are available only for specific types of replication or publishing configurations.

The replication programs store information about your configurations in control tables. The ASNCLP commands create, modify, and remove this information.

For example, the ASNCLP program provides a command to create a Q subscription. In Q replication, the output of a command is a SQL script that inserts information into the control tables about the source, target, queues, and other options.

Each ASNCLP command works with only a subset of information. You can use multiple commands together to generate the SQL for an entire configuration.

Before you run the ASNCLP program, you need to configure your environment. Once your environment is configured, you’ll need to know a few basics about using the ASNCLP program.

Supported operating systems

The ASNCLP program runs where ever DB2® for Linux®, UNIX®, and Windows® is installed. The ASNCLP program does not run natively on z/OS® or System i™.

The ASNCLP commands will generate replication definitions for all operating system environments that are supported by the replication products: z/OS, System i (SQL replication only), Linux, UNIX, and Windows. You must have connectivity to each server for which you are generating replication definitions; that is, you must be able to issue a db2 connect statement to each of the servers.

Restriction: The ASNCLP program does not support z/VM® or VSE because DB2 in these operating-system environments does not support the replication architecture for DB2 Version 8 and later.

Setting up a Java environment to run the ASNCLP program

The ASNCLP program runs in a Java environment. Your PATH environment variable must contain a path to a Java runtime environment in order to run the ASNCLP.

About this task

Use the following procedure if the PATH environment does not contain a path to a Java runtime environment.

Procedure

To set up a Java environment to run the ASNCLP program, add the following path to your PATH environment variable:
Where \texttt{INSTDIR} is the DB2 instance directory. On Linux and UNIX, the instance directory is the \texttt{INSTDIR/sqllib} directory, where \texttt{INSTDIR} is the home directory of the instance owner. On Windows, the instance directory is the \texttt{\sqllib} directory where DB2 was installed.

**Examples**

- **Linux/UNIX** To set the PATH environment variable from a UNIX command prompt:
  ```bash
  export CLASSPATH=$PATH:/u/myinst/sqllib/java/jdk;
  ```

- **Windows** To set the PATH environment variable from a Windows command prompt:
  ```batch
  set CLASSPATH=%PATH%;%c:\Program Files\sqllib\java\jdk;
  ```

**Binding z/OS packages for the ASNCLP program**

Before you use the ASNCLP program with DB2 for z/OS, you must bind the basic DRDA and CLI packages to the DB2 subsystem that you will be working with.

**Before you begin**

Before you can bind the z/OS packages, you must connect to the DB2 subsystem on the z/OS server.

**Procedure**

To bind the basic z/OS packages for the ASNCLP program, open an operating system command prompt and issue the following command:

```batch
bind @ddcsmsv.lst blocking all sqlerror continue
db2 bind @db2cli.lst isolation ur blocking all
```

If you do not perform this bind, the first time you use the ASNCLP program with a DB2 for z/OS server, the ASNCLP program might return the following error message:

```
ASN1560E The replication action ended in error. An SQL error was encountered.
SQL Message: "[IBM][CLI Driver][DB2] SQL0805N Package "package_name" was not found. SQLSTATE=51002
```

**Running the ASNCLP commands in interactive mode**

You can run the ASNCLP commands in interactive mode from a command prompt.

**Procedure**

To run the ASNCLP commands in interactive mode:

1. Open an operating system command prompt and issue the following command:
   ```batch
   ASNCLP
   ```
The ASNCLP command starts the ASNCLP program and changes the command prompt to Repl >.

2. Issue any of the ASNCLP commands. For example: To set the Q Capture server to the database aliasname, issue the following command:
   SET SERVER CAPTURE TO DBALIAS aliasname

3. To exit the ASNCLP program, issue the following command:
   quit

To get help for the ASNCLP program, issue the following command from an operating system command prompt:
   ASNCLP ?

---

### Running the ASNCLP commands in execute-immediately mode

The execute-immediately mode is useful when you need to issue a single command. You can use the START QSUB command, STOP QSUB command, and LIST commands in execute-immediately mode.

**Before you begin**

The ASNCLP command that you execute cannot rely on previous commands. The command must be self-contained. For example, many commands rely on the SET SERVER command to define where objects are created.

**Procedure**

To execute an ASNCLP command in execute-immediately mode:

1. Open an operating system command prompt.
2. Run the ASNCLP command:
   
   ```
   ASNCLP -exe my_command
   ```

   Replace `my_command` with the ASNCLP command that you want to immediately execute.

   The following command is an example of starting a Q subscription for a Classic replication source:
   
   ```
   asnclp -exe START QSUB SUBNAME sub1 CAP SERVER OPTIONS CONFIG SERVER classic1 FILE asnservers.ini ID id1 PASSWORD passwd1
   ```

---

### Running the ASNCLP commands using an input file

You can run the ASNCLP commands in batch mode by using an input file.

**Procedure**

To run the ASNCLP commands in batch mode using an input file:

1. Create an input file that contains the ASNCLP commands that you want to run. Commands in the input file must be delimited by the semicolon (;) and can span multiple lines. You can also add comments to the input file by beginning the comment line with a number (#) sign.
2. Open an operating system command prompt and issue the following command:
   
   ```
   ASNCLP -f myfile.in
   ```
In the example the input-file name is myfile.in and can consist of any valid file name plus an extension. You can also specify a full file path and file name. For example:

```
ASNCLP -f c:\temp\myfile.in
```

The ASNCLP command starts the ASNCLP program, which processes all of the commands in the input file until it encounters an error or the end of the file.

If your input file does not contain the quit command, you can exit the ASNCLP program by issuing the following command:

```
quit
```

### ASNCLP configuration file

The ASNCLP command-line interface requires connectivity information be provided so that it can access Classic replication sources. The connection information is provided through a configuration file.

The ASNCLP configuration file contains a group of lines for each Classic replication source that the ASNCLP needs to access. Each grouping has a unique name for the group followed by lines that specify the connection information. The unique name is used in ASNCLP scripts to identify a source.

**Syntax**

Specify the server information in the configuration file in the following format:

```
[NAME]
Type=source_type
Data source=data_source_name
Host=host_name
Port=port_number
Codepage=code_page
...
```

**Parameters**

**NAME**

Specifies a unique name for a configuration. You provide this name in ASNCLP scripts so that the ASNCLP command-line program can connect to a Classic replication source. You can define multiple servers in a single configuration file by indicating the beginning of a new server definition in the enclosing brackets (for example, `NAME2`).

**Important:** The value cannot be longer than eight characters.

**Type**

Specifies the type of server. For Classic replication, specify `Type=classic`.

**Data source**

Specifies the name of the query processor on the Classic data server.

**Host**

Specifies the host name or IP address of the data server where the `data_source_name` resides.

**Port**

Port is the port number of the Classic server where the data source resides.
Codepage

Codepage is an optional parameter that describes the codepage of the data in the Classic data server.

Example

The following example shows a configuration file with two server definitions:

```
[server_1]
Type=classic
Data source=CACSAMP1
Host=123.123.123.1
Port=8096
[server_2]
Type=classic
Data source=CACSAMP2
Host=123.123.231.5
Port=9016
```

Usage notes

You can save the configuration file to any location. The recommended file name is asnservers.ini.

You must use the SET SERVER command to provide the ASNCLP program with the location of the configuration file. The following example shows that the asnservers.ini configuration file is saved in the /home/db2inst/sqlib/classic_files directory.

```
SET SERVER capture TO CONFIG SERVER cacsamp1 FILE "/home/db2inst/sqlib/classic_files/asnservers.ini" ID my_user_id
PASSWORD "my_password";
```
Chapter 2. ASNCLP commands for SQL replication

The ASNCLP commands for SQL replication define and change objects such as control tables, registrations, and subscription sets.

“Sample ASNCLP scripts for setting up SQL replication” on page 8 demonstrates how you can combine SQL replication commands to create an ASNCLP setup script.

Table 1 lists the ASNCLP commands for SQL replication and links to topics that describe each command.

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add columns to an existing member</td>
<td>“ALTER MEMBER ADD COLS command” on page 11</td>
</tr>
<tr>
<td>Change the properties of a registration</td>
<td>“ALTER REGISTRATION command” on page 12</td>
</tr>
<tr>
<td>Change the properties of a subscription set</td>
<td>“ALTER SUBSCRIPTION SET command” on page 15</td>
</tr>
<tr>
<td>Establish a session for SQL replication</td>
<td>“ASNCLP SESSION SET TO command (SQL replication)” on page 17</td>
</tr>
<tr>
<td>Create control tables</td>
<td>“CREATE CONTROL TABLES command (SQL replication)” on page 18</td>
</tr>
<tr>
<td>Create a subscription-set member</td>
<td>“CREATE MEMBER command” on page 20</td>
</tr>
<tr>
<td>Create a registration</td>
<td>“CREATE REGISTRATION command” on page 28</td>
</tr>
<tr>
<td>Create a SQL statement that is processed with an existing subscription set</td>
<td>“CREATE STMT command” on page 32</td>
</tr>
<tr>
<td>Create a subscription set</td>
<td>“CREATE SUBSCRIPTION SET command” on page 34</td>
</tr>
<tr>
<td>Drop control tables</td>
<td>“DROP CONTROL TABLES command” on page 36</td>
</tr>
<tr>
<td>Delete a subscription-set member</td>
<td>“DROP MEMBER command” on page 37</td>
</tr>
<tr>
<td>Delete a registration</td>
<td>“DROP REGISTRATION command” on page 38</td>
</tr>
<tr>
<td>Delete SQL statements for an existing subscription set</td>
<td>“DROP STMT command” on page 38</td>
</tr>
<tr>
<td>Delete a subscription set</td>
<td>“DROP SUBSCRIPTION SET command” on page 39</td>
</tr>
<tr>
<td>Control a manual full refresh for offline load procedures</td>
<td>“OFFLINE LOAD command” on page 40</td>
</tr>
<tr>
<td>Promote a registration</td>
<td>“PROMOTE REGISTRATION command” on page 40</td>
</tr>
<tr>
<td>Promote a subscription set</td>
<td>“PROMOTE SUBSCRIPTION SET command” on page 42</td>
</tr>
<tr>
<td>Set a source and target Capture schema for all task commands</td>
<td>“SET CAPTURE SCHEMA command (SQL replication)” on page 44</td>
</tr>
<tr>
<td>Specify whether to drop the table space when you drop the replication object that it contains</td>
<td>“SET DROP command (SQL replication)” on page 45</td>
</tr>
<tr>
<td>Set the log file name for the ASNCLP program</td>
<td>“SET LOG command” on page 46</td>
</tr>
<tr>
<td>Specify a name for the output files that contain the SQL scripts</td>
<td>“SET OUTPUT command (SQL replication)” on page 47</td>
</tr>
<tr>
<td>Set up customization rules for creating table space objects</td>
<td>“SET PROFILE command (SQL replication)” on page 57</td>
</tr>
<tr>
<td>Specify whether to automatically run the SQL statements before the ASNCLP commands process the next task command</td>
<td>“SET RUN SCRIPT command (SQL replication)” on page 51</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 2004, 2007
Table 1. ASNCLP commands for SQL replication (continued)

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify the server (database) used in the ASNCLP session, authentication information, and other required parameters for connecting to the server</td>
<td>“SET SERVER command (SQL replication)” on page 54</td>
</tr>
<tr>
<td>Enable and disable the tracing for the ASNCLP commands</td>
<td>“SET TRACE command” on page 56</td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up SQL replication

This sample contains five ASNCLP scripts for setting up a basic SQL replication environment. It includes Capture control tables, a registration, Apply control tables, a subscription set, and a subscription-set member.

Each ASNCLP script generates one or more SQL scripts to create a replication object. Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:

1. Capture control tables
2. Registration
3. Apply control tables
4. Subscription set (generates two SQL scripts that create definitions at the Capture control server and the Apply control server).
5. Subscription-set member (generates two SQL scripts that create definitions at the Capture control server and the Apply control server, which is also the target server).

Table 2 on page 11 below the sample describes each SQL script.

This sample has a section for each ASNCLP script, which you can copy to a text file and run by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

ASNCLP script 1 (Capture control tables)

This script generates SQL statements that create Capture control tables at the SAMPLE database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Capture control tables
3. Ending the ASNCLP session

```sql
# 1 Setting the environment
# The scope of environment commands is the entire ASNCLP script file unless
# the commands are overridden by another value.
# In the SET SERVER command, the user ID and password are optional. If you omit
# these keywords, the ASNCLP will use the implicit ID and password for connecting
# to the database.
# The SET OUTPUT command generates a SQL script, capctrl.sql.
# The SET LOG command sets one log file, capctrl.err, to record results for the
# ASNCLP script.
# The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
# they are run.
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET OUTPUT CAPTURE SCRIPT "capctrl.sql";
```
SET LOG "capctrl.err";
SET RUN SCRIPT LATER;

# 2 Creating Capture control tables
# This command generates a SQL script that connects to the SAMPLE database and
# creates the control tables. The UW table will be created in the TSUOW100
# table space, and the other control tables will be created in the TSASN100
# table space.

CREATE CONTROL TABLES FOR CAPTURE SERVER IN UW UW OW TSUOW100 OTHERS TSASN100;

# 3 Ending the ASNCLP session.
QUIT;

**ASNCLP script 2 (registration)**

This script generates SQL statements that register the STAFF table at the SAMPLE database for replication. It includes commands for the following tasks:

1. Setting the environment
2. Registering a source table
3. Ending the ASNCLP session.

# 1 Setting the environment
# The SET OUTPUT command generates a SQL script, register.sql.

SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET OUTPUT CAPTURE SCRIPT "register.sql";
SET LOG "register.err";
SET RUN SCRIPT LATER;

# 2 Registering a source table.
# The CREATE REGISTRATION command specifies the STAFF table. The
# DIFFERENTIAL REFRESH STAGE keywords specify to update the target table
# periodically when the source table changes, and create a change-date (CD)
# table named CDSTAFF. Because the COLS keyword is not specified, all columns
# in the source table are registered.

CREATE REGISTRATION (DB2ADMIN.STAFF) DIFFERENTIAL REFRESH STAGE CDSTAFF;

# 3 Ending the ASNCLP session.
QUIT;

**ASNCLP script 3 (Apply control tables)**

This script generates SQL statements that create the Apply control tables at the TARGET database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Apply control tables
3. Ending the ASNCLP session.

# 1 Setting the environment
# The SET OUTPUT command generates a SQL script, appctrl.sql.

SET SERVER CONTROL TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET OUTPUT CONTROL SCRIPT "appctrl.sql";
SET LOG "appctrl.err";
SET RUN SCRIPT LATER;

# 2 Creating Apply control tables
# These statements generate a SQL script that connects to the TARGET database
# and creates the control tables.

CREATE CONTROL TABLES FOR APPLY CONTROL SERVER IN UW OTHERS TSASN100;
# 3 Ending the ASNCLP session.

QUIT;

**ASNCLP script 4 (subscription set)**

This script generates SQL statements that create a subscription set. It includes commands for the following tasks:

1. Setting the environment
2. Creating a subscription set
3. Ending the ASNCLP session.

# 1 Setting the environment
# Two SET SERVER commands are used because the subscription set definitions are stored in both the Capture control tables and Apply control tables.
# The SET OUTPUT command generates two scripts: capsubset.sql, which inserts set definitions into the Capture control tables, and appsubset.sql, which inserts set definitions into the Apply control tables.

```
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER CONTROL TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET OUTPUT CAPTURE SCRIPT "capsubset.sql" CONTROL SCRIPT "appsubset.sql";
SET LOG "subset.err";
SET RUN SCRIPT LATER;
```

# 2 Creating a subscription set
# The CREATE SUBSCRIPTION SET command creates a set named SET00 with an Apply qualifier of AQ00. It specifies that the set be activated indefinitely on a specified start date and time, and that the set be processed every minute.

```
CREATE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 ACTIVATE YES
TIMING INTERVAL 1 START DATE "2006-10-22" TIME "09:00:00.000000";
```

# 3 Ending the ASNCLP session.

QUIT;

**ASNCLP script 5 (subscription-set member)**

This script generates SQL statements that create a subscription-set member. It includes commands for the following tasks:

1. Setting the environment
2. Creating a subscription-set member
3. Ending the ASNCLP session.

# 1 Setting the environment
# The SET OUTPUT command generates two SQL scripts, capmember.sql and appmember.sql. Because the target server is the same as the Apply control server, no separate script is generated for the target server. These scripts define the member at the Capture control server and Apply control server.

```
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER CONTROL TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET OUTPUT CAPTURE SCRIPT "capmember.sql" CONTROL SCRIPT "appmember.sql"
SET LOG "member.err";
SET RUN SCRIPT LATER;
```

# 2 Creating a subscription-set member.
# The SET PROFILE command creates a profile, TBSPROFILE, to store options for the tablespace that is used by the target table. The CREATE MEMBER command
# specifies the SET00 set, AQ00 Apply qualifier, and STAFF source table. The
# TRGSTAFF target table is specified as a user copy with all columns registered.

SET PROFILE TBSPROFILE FOR OBJECT TARGET TABLESPACE OPTIONS UW USING
FILE "/tmp/db/ts/TSTRG.TS" SIZE 700 PAGES;
CREATE MEMBER IN SETNAME SET00 APPLYQUAL AQ00 ACTIVATE YES SOURCE STAFF
TARGET NAME TRGSTAFF DEFINITION IN TSTRG00 CREATE USING PROFILE TBSPROFILE
TYPE USRCOPY COLS ALL REGISTERED;

# Ending the ASNCLP session.
QUIT;

Output of the scripts

In addition to the log files, this sample produces six SQL script files in the same
directory where you run the ASNCLP program. Table 2 describes the files.

Table 2. SQL script files that are created by the sample ASNCLP scripts

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>capctrl.sql</td>
<td>Create Capture control tables</td>
</tr>
<tr>
<td>register.sql</td>
<td>Register a source table</td>
</tr>
<tr>
<td>appctrl.sql</td>
<td>Create Apply control tables</td>
</tr>
<tr>
<td>capsubset.sql</td>
<td>Insert definitions for a subscription set into the Capture control tables</td>
</tr>
<tr>
<td>appsubset.sql</td>
<td>Insert definitions for a subscription set into the Apply control tables</td>
</tr>
<tr>
<td>capmember.sql</td>
<td>Insert definitions for a subscription-set member into the Capture control tables</td>
</tr>
<tr>
<td>appmember.sql</td>
<td>Insert definitions for a subscription-set member into the Apply control tables</td>
</tr>
</tbody>
</table>

**ALTER MEMBER ADD COLS command**

Use the ALTER MEMBER ADD COLS command to add columns to an existing member in an existing subscription set.

**Syntax**

```
ALTER MEMBER ADD COLS IN SETNAME setname APPLYQUAL applyqual SOURCE
                   objowner objname TARGET objowner objname
                   COLS EXPRESSION "source-col-or-expr"
```

**Parameters**

**SETNAME setname**

Specifies the subscription-set name.

**APPLYQUAL applyqual**

Specifies the Apply qualifier for the subscription set.
**SOURCE objowner.objname**
Specifies the source object's owner and name.

**TARGET objowner.objname**
Specifies the target object's owner and name.

**COLS**
Specifies the columns to add. You can specify multiple columns by using commas and parentheses.

**EXPRESSION "source-col-or-exp"**
Specifies an expression for the column. The double quotation marks are required.

**TARGET name**
Specifies the target’s column name.

+ Specifies that the column is part of the primary key.

**Usage notes**
- For update-anywhere subscription sets, the columns are added to the members for both replication directions (master-to-replica and replica-to-master).
- The Capture schema for the target table is inherited from the subscription set.

**Example**
To add column NEWSTAFF to the existing subscription set SET00:

```
ALTER MEMBER ADD COLS IN SETNAME SET00 APPLYQUAL AQ00 SOURCE DB2ADMIN.STAFF
TARGET DB2ADMIN.TRGSTAFF COLS (EXPRESSION "source-col-or-exp" TARGET NEWSTAFF)
```

**ALTER REGISTRATION command**
Use the ALTER REGISTRATION command to alter a registration row in the IBMSNAP_REGISTER table and to add new columns to a registered source.

**Syntax**
```
ALTER REGISTRATION
   ROW row-clause
      ADD
         add-cols-clause
```

**row-clause:**
```
   (objowner.objname)
   CONFLICT NONE STANDARD ENHANCED
   UPDATE AS DELETE INSERT OFF ON
   CAPTURE ALL CHANGES
   FORWARDING OFF ON
```
Parameters

ROW
Specify to alter a registration row in the IBMSNAP_REGISTER table.

ADD
Specify to add new columns from a source object to a registration. This parameter only applies if the source object is a table or nickname.

$objowner$
Specifies the owner of the registered source object (table, view, or nickname). You can specify multiple objects.

$objname$
Specifies the name of the registered source object (table, view, or nickname). You can specify multiple objects.

CONFLICT
Specifies the conflict-detection level.

NONE
No conflict detection. Conflicting updates between the master table and the replica table will not be detected. This option is not recommended for update-anywhere replication. This is the default.

STANDARD
Moderate conflict detection. During each Apply cycle, the Apply program compares the key values in the master’s CD table with those in the replica’s CD table. If the same key value exists in both CD tables, it is a conflict. In the case of a conflict, the Apply program will undo the transaction that was previously committed at the replica by reading from the replica’s CD table and keeping only the changes that originated at the master.

ENHANCED
Conflict detection that provides the best data integrity among the master and its replicas. As with standard detection, the Apply program compares the key values in the master’s CD table with those in the replica’s CD table during each Apply cycle. If the same key value exists in both CD tables, it
is a conflict. However, with enhanced detection, the Apply program waits for all in-flight transactions to commit before checking for conflicts. To ensure that it catches all in-flight transactions, the Apply program locks all target tables in the subscription set against further transactions and begins conflict detection after all changes are captured in the CD table. In case of a conflict, the Apply program will undo the transaction that was previously committed at the replica by reading from the replica’s CD table and keeping only the changes that originated at the master.

**UPDATE AS DELETE INSERT**

**ON**
Specify to capture updates as delete-insert pairs.

**OFF**
Specify to capture updates as updates. This is the default.

**CAPTURE**

**ALL**
Specify to capture everything.

**CHANGES**
Specify to capture only changes.

**FORWARDING**

**OFF**
Specify not to forward changes from this source.

**ON**
Specify to forward changes from this source.

**FULL REFRESH**

**ON**
Specify to allow full refreshes for this source.

**OFF**
Specify to not allow full refreshes for this source.

**STOP ON ERROR**

**ON**
Specify to stop the Capture program if it detects an error for this registration.

**OFF**
Specify to not stop the Capture program if it detects an error for this registration.

**COLS**
Specifies the columns that you want to register.

`colname`
Specifies a list of the columns that you want to register.

**IMAGE**

**AFTER**
Specify to register only after-image columns.

**BOTH**
Specify to register both after-image and before-image columns.
BEFORE
Specify to register only before-image columns.

PREFIX
- If you specify IMAGE AFTER, the prefix will be null and the source will not allow any before-image columns.
- If you specify IMAGE BOTH or IMAGE BEFORE and do not specify PREFIX, a default value of X is used as a prefix for the before images. If you specify PREFIX, that value is used.
- If you choose IMAGE BOTH and do not specify a prefix, the before-imaged prefix will be X.

You cannot alter an existing before-image prefix using the ALTER REGISTRATION ROW command. However, you can add that prefix to a new before-image column. If the existing before-image prefix is null and you want to add a before-image column to the existing registration, you can specify the before-image prefix using the ALTER REGISTRATION ADD command. If you do not specify the prefix, the ASNCLP program sets it to a default value of X.

Usage notes
The parameters in this command do not have default values.

If you add a column to a CD table when the registered source also has an internal CCD table associated with it, you must:
- Use the ALTER ADD REGISTRATION COL command to add a column to the CD table
- Use the ALTER ADD SUBSCRIPTION MEMBER COL command to add a column to the internal CCD table. If you do not do this step, you will not be able to add the column to any target table that is dependent on the registered source.

Example 1
To alter a registration row for DB2ADMIN.STAFF that captures updates as delete-insert pairs:
ALTER REGISTRATION ROW (DB2ADMIN.STAFF) UPDATE AS DELETE INSERT ON

Example 2
To alter a registration by adding a new column C002 to table DB2ADMIN.STAFF:
ALTER REGISTRATION ADD DB2ADMIN.STAFF COLS (C002 IMAGE BOTH)

ALTER SUBSCRIPTION SET command
Use the ALTER SUBSCRIPTION SET command to alter certain values for a subscription set.

Syntax

```
ALTER SUBSCRIPTION SET SETNAME setname APPLYQUAL applyqual SETTYPE
```
Parameters

**SETNAME** *setname*
Specifies the subscription-set name.

**APPLYQUAL** *applyqual*
Specifies the Apply qualifier for the subscription set.

**SETTYPE**
Specifies the subscription set type.

- **R**
  Specifies a read-only set. This is the default.

- **U**
  Specifies an update-anywhere set. The default is both F and S directions.
    - **F ONLY**
      Specifies an update-anywhere set in the F direction only, where the source table is the replica and the target table is the master.
    - **S ONLY**
      Specifies an update-anywhere set in the S direction only, where the source table is the mast table or the other source, and the target table is the replica or other copy.

- **P**
  Specifies a peer-to-peer set.

**ACTIVATE**
Specifies whether to activate the subscription set.

- **NO**
  Specify to not activate the subscription set. This is the default.

- **YES**
  Specify to activate the subscription set.

- **ONCE**
  Specify to activate the subscription set for one Apply cycle, then deactivate the subscription set.

**TIMING**
Specifies the timing for the subscription set.

- **EVENT** *eventname*
  Specifies the event that when posted to the IBMSNAP_SUBS_EVENT table, causes the Apply program to process the subscription set.

- **INTERVAL** *minutes*
  Specifies the interval for the Apply program to process the subscription set. The default interval is 20 minutes.

- **BOTH**
  Specifies that this subscription set uses both event and interval timing.
CONTINUOUS
   Specifies that the Apply program should process the subscription set continuously. This keyword is equivalent to specifying an interval of zero minutes.

BLOCKING minutes
   Specifies a threshold limit to regulate the amount of data to fetch and apply. This keyword controls the MAX_SYNCH_MINUTES column of the IBMSNAP_SUB_SET table.

COMMIT COUNT n
   Specifies the number of transactions that the Apply program should process before issuing a SQL COMMIT statement for the subscription set. Specify a NULL value to have the Apply program issue just one COMMIT statement for the subscription set after it processes the entire set.

Example 1

To alter the SET00 subscription set to a read-only subscription set type using the AQ00 Apply qualifier and to change the timing interval from 20 minutes to 15 minutes:
ALTER SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 SETTYPE R
   ACTIVATE YES TIMING INTERVAL 15 COMMIT COUNT NULL

Example 2

To alter the SET00 subscription set so that it activates once and sets the source table as the replica and the target table as the master:
ALTER SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 SETTYPE U
   F ONLY ACTIVATE ONCE COMMIT COUNT 5

ASNCLP SESSION SET TO command (SQL replication)

Use the ASNCLP SESSION SET TO command to define an ASNCLP session for SQL replication.

Syntax

ASNCLP SESSION SET TO—SQL REPPLICATION

Parameters

SQL REPLICATION
   Specify to set the ASNCLP session to SQL replication. This ASNCLP session only accepts SQL replication syntax.

Usage notes

Issue the ASNCLP SESSION SET command before all other commands in an ASNCLP session. If you do not issue the ASNCLP SESSION SET command, the ASNCLP program defaults to SQL replication.

Example

To set the ASNCLP session to SQL replication:
ASNCLP SESSION SET TO SQL REPPLICATION
CREATE CONTROL TABLES FOR command (SQL replication)

Use the CREATE CONTROL TABLES FOR command to create a new set of Capture, Apply, or Replication Alert Monitor control tables.

**Syntax**

```
CREATE CONTROL TABLES FOR CAPTURE SERVER
    APPLY CONTROL SERVER
    MONITOR CONTROL SERVER
    IN ZOS zos-ts-clause
    UW uw-ts-clause
    NONIBM federated-clause
```

zos-ts-clause:

```
UOW DB dbname tsname
    NAMING PREFIX prefix
    prof-clause
```

ALERTS DB dbname tsname
    NAMING PREFIX prefix
    prof-clause

PAGE LOCK DB dbname tsname
    NAMING PREFIX prefix
    prof-clause

ROW LOCK DB dbname tsname
    NAMING PREFIX prefix
    prof-clause

uw-ts-clause:

```
UOW tsname
    NAMING PREFIX prefix
    prof-clause
```

OTHERS tsname
    NAMING PREFIX prefix
    prof-clause

federated-clause:

```
OTHERS tsname
    NAMING PREFIX prefix
    prof-clause
```

```
SCHEMA schemaname
```

prof-clause:

```
CREATE USING PROFILE pname
    REUSE
```
Parameters

CAPTURE SERVER
Specify to create replication control tables for the Capture server.

APPLY CONTROL SERVER
Specify to create replication control tables for the Apply control server.

MONITOR CONTROL SERVER
Specify to create replication control tables for the Monitor control server.

IN
Specifies the table space. If you do not specify the IN clause, the CREATE CONTROL TABLES command uses the DB2 defaults for table spaces.

ZOS
Specifies z/OS or OS/390®.

UW
Specifies Linux, UNIX, or Windows.

NONIBM
Specifies federated data source such as Oracle or Informix®.

Federated-clause
OTHERS
Specifies the table space for all replication control tables whenever the tables are created in a non-DB2 database. You specify a table space name or a segment name for only those remote sources that support them.

SCHEMA
Specifies the remote schema name for a federated replication source server. The default is the remote user ID. If the schema is in lower or mixed case on the federated data source, you must use double quotation marks around the string to ensure that it is not converted to upper case. Lower case names and quotation marks are recommended for Informix sources.

UOW
Specifies the table space for the unit-of-work (UOW) table.

ALERTS
Specifies an existing database on z/OS to create the control tables in. This keyword is valid only when creating monitor control servers.

PAGE LOCK
Specifies the table space for replication control tables that require page-level locking. The table must be in an existing database.

ROW LOCK
Specifies the table space for replication control tables that require row-level locking. The table must be in an existing database.

DB dbname
Specifies the name of an existing database. You must specify the database name, even if you set the database name in the profile.

OTHERS
Specifies the table space for all replication control tables except the UOW table.

tsname
Specifies the table space name for the monitor alerts table. The tsname input can be a heterogeneous segment or table space name.
NAMING PREFIX `prefix`
Specifies a naming prefix for the control tables.

CREATE USING PROFILE `pname`
Specify to create the control tables using the `pname` profile. If you specify the
CREATE USING PROFILE parameter, the ASNCLP program uses `tsname` as
the key (for z/OS, the key is `dbname.tsname`).

REUSE
Specify to reuse the current table space or index. You must issue the CREATE
USING PROFILE parameter before you can use the REUSE parameter. When
you specify the REUSE parameter, the ASNCLP program checks if the table
space or index exists for the `tsname`:
• If the table space or index exists, the ASNCLP program resets the flags and
  passes the fully populated object.
• If the table space or index does not exist, the ASNCLP program displays a
  syntax error saying that the CREATE USING PROFILE parameter is
  expected.

Example 1
To create the Capture control tables and to name the UOW table space TSUOW100
and all other table spaces TSASN100:
CREATE CONTROL TABLES FOR CAPTURE SERVER IN UW UOW TSUOW100 OTHERS TSASN100

Example 2
To create the Apply control tables and to name all table spaces except the UOW
table space TSASN100:
CREATE CONTROL TABLES FOR APPLY CONTROL SERVER IN UW OTHERS TSASN100

CREATE MEMBER command
Use the CREATE MEMBER command to add a subscription-set member to an
existing subscription set.

Adding a member to a set includes:
• Creating the mapping between the source and target tables (database objects).
• Creating the mapping between the source and target columns.
• Creating the target table (database object), if it doesn’t already exist.
• Creating the target index, if necessary.
• Setting the KEYS value for the index.

Syntax
```
CREATE MEMBER IN SETNAME `setname` APPLYQUAL `applyq` ACTIVATE
SOURCE

TGT KEY CHANGE

OFF

ON

objname

TARGET-clause
```
Parameters

**SETNAME** *setname*
Specifies the subscription-set name.

**APPLYQUAL** *applyqual*
Specifies the Apply qualifier for the subscription set.

**ACTIVATE**
Specifies whether to activate the subscription set.
NO
Specify to not activate the subscription set. This is the default.

YES
Specify to activate the subscription set.

ONCE
Specify to activate the subscription set for one Apply cycle, then deactivate the subscription set.

SOURCE objowner.objname
Specifies the source object name and owner.

TGT KEY CHANGE
Specifies whether the target key can change.

OFF
Specifies that the key value cannot change. This is the default.

ON
Specifies that the key value can change.

WHERE "sql-where-stmts"
Specifies the WHERE clause that will be evaluated for this member. The double quotation marks are required.

COLS
Specifies the columns to include in the target table.

ALL REGISTERED
Specify to include all registered columns.

INCLUDE
Specifies the columns to include.

EXPRESSION "expr"
Specifies the source column or expression. Specify multiple columns or expressions using commas and parentheses.

TARGET name
Specifies the name of the target column.

EXCLUDE (colname)
Exclude the specified columns.

KEYS keyname
Specifies the key names. Include a plus sign (+) for ascending keys and a minus sign (-) for descending keys.

target-clause:

TARGET
Specifies the target object.

NAME owner.name
Specifies the target object owner and name.

NAMING PREFIX prefix
Specifies the prefix to use to generate a target-table name.

DEFINITION
Specifies the database, table space, and target-table type.

trg-def-clause:
IN Specifies the table space for the target table. If you do not specify the IN clause, the command uses the DB2 defaults for table spaces.

DB name
   Specifies the name of the database that contains the target table and its table space. You must specify the database name, even if you set the database name in the profile.

tsname
   Specifies the name of the table space. For z/OS, the name includes the database name (for example, "dbname.tsname"). This command does not create the database. You can specify a heterogeneous segment or table space name, but it must already exist.

NAMING PREFIX prefix
   Specifies a naming prefix to use to create the table space.

TYPE
   Specifies the type of target table.

PIT
   Specifies a point-in-time table.

USERCOPY
   Specifies a user-copy table.

BASEAGGREGATE
   Specifies a base-aggregate table. This table contains data aggregated from the source or point-in-time table at intervals.

CHANGEAGGREGATE
   Specifies a change-aggregate table. This table contains data based on changes to a source table (CD or internal CCD table).

REPLICA
   Specifies a replica table for update-anywhere replication.

CCD
   Specifies a consistent-change data (CCD) table.

EXTERNAL
   Specifies that the CCD table is external.

INTERNAL
   Specifies that the CCD table is internal.

prof-clause:

CREATE USING PROFILE pname
   Specify to use the tsname value as the key (for z/OS, the key is dbname.tsname).

REUSE
   Specify to reuse the current table space or index. You must issue the CREATE USING PROFILE parameter before you can use the REUSE parameter. When you specify the REUSE parameter, the ASNCLP program checks if the table space or index exists for the tsname:
   • If the table space or index exists, the ASNCLP program resets the flags and passes the fully populated object to the API.
   • If the table space or index does not exist, the ASNCLP program displays a syntax error saying that the CREATE USING PROFILE parameter is expected.
replica-clause:

**CD cdowner.cdname**

Specifies the name of the object owner and the name of the CD table for the replica table.

**UPDATE AS DELETE INSERT**

Specifies how to handle SQL UPDATE statements.

- **OFF**
  - Specify to capture updates as updates. This is the default.
- **ON**
  - Specify to capture updates as delete-insert pairs.

**FORWARDING**

Specifies whether to forward captured changes to other replicas.

- **OFF**
  - Specify to not forward captured changed.
- **ON**
  - Specify to forward captured changes.

**FULL REFRESH**

Specifies whether to perform a full refresh for the replica table.

- **ON**
  - Specify to perform a full refresh. This is the default.
- **OFF**
  - Specify not to perform a full refresh.

**STOP ON ERROR**

Specifies whether the Capture program is to stop when it encounters an error.

- **ON**
  - Specify to stop the Capture program if a Capture error occurs. This is the default.
- **OFF**
  - Specify to continue the Capture program if a Capture error occurs.

ccd-clause:

join-options:

**JOIN CD UOW**

Specifies that the CD table and IBMSNAP_UOW table are joined to obtain commit information for transactions. The CCD table is created as type 3.

**AS SOURCE**

Specifies that the CCD table is a source.

**WITH UOW COLS**

- **ALL**
  - Specifies that the CCD table includes columns from the IBMSNAP_UOW table.
- **COMPLETE**
  - Specifies whether the CCD table is complete.
  - **ON**
    - Specifies that the CCD table includes all data. This is the default.
OFF
Specifies that the CCD table includes only changes.

CONDENSED
Specifies whether to condense the CCD table.

ON
Specifies that the CCD table includes only the most recent change for each row. This is the default.

OFF
Specifies that the CCD table includes a change history for each row.

cols-clause:

colname
Specifies which of the UOW columns should be included in the CCD table.
These columns include: IBMSNAP_APPLY_QUAL, IBMSNAP_AUTHID, IBMSNAP_AUTHTKN, IBMSNAP_REJ_CODE, and IBMSNAP_UOWID.

no-join-options:

NO JOIN CD UOW
Specifies that you do not want the CD table and IBMSNAP_UOW table to be joined. The CCD table will be created with type 9.

AS SOURCE
Specifies that the CCD table is a source.

COMPLETE
Specifies whether the CCD table is complete.

ON
Specifies that the CCD table includes all data. This is the default.

OFF
Specifies that the CCD table includes only changes.

CONDENSED
Specifies whether to condense the CCD table.

ON
Specifies that the CCD table includes only the most recent change for each row. This is the default.

OFF
Specifies that the CCD table includes a change history for each row.

loadx-clause:

LOADX TYPE
Specifies the load type to use with this member.

NO ASNLOAD
Specify to not use the ASNLOAD for this member.

USER DEFINED
Specify to use a user-defined or user-modified ASNLOAD exit.

CROSSLOADER LOAD SRC NICKNAME owner.tablename
Specify the owner and tablename to use with the LOAD from CURSOR utility for this member.
LOAD EXPORT
Specify to use an EXPORT/LOAD combination for this member.

IMPORT EXPORT
Specify to use an EXPORT/IMPORT combination for this member.

Usage notes
- The target object is not required for the command, but the command does require a target object so that the ASNCLP program can derive the target name.
- You cannot specify the conflict-detection level for replica-table autoregistration because it is inherited from the master table.
- You cannot specify capturing updates as delete-insert pairs for CCD table autoregistration because there is no Capture program for these tables.
- If the subscription set is empty when you issue this command, the command uses a default value of YES for the ACTIVATE keyword.

Example 1
In this example, you create a member in the SET00 subscription set for mapping the STAFF source table to the TRGSTAFF target table. The TRGSTAFF table is created in the TSUOW100 table space and the index for the TRGSTAFF table is created according to the settings in the TBSPROFILE profile.

```
CREATE MEMBER IN SETNAME SET00 APPLYQUAL AQP00 SOURCE DB2ADMIN.STAFF
   TARGET NAME DB2ADMIN.TRGSTAFF DEFINITION IN TSUOW100 CREATE USING
   PROFILE TBSPROFILE;
```

Example 2
The following commands set the environment and create a subscription set member with a Linux, UNIX, or Windows database as the Capture server and a Microsoft® SQL Server target. The Apply control server is the same as the Capture server. The member has the following attributes:
- Subscription set name: SET1
- Apply qualifier: APPQUAL1
- Source owner: repldba
- Source table: EMPLOYEE
- Target nickname owner: repldba
- Target nickname: TRGEMPNICK

The commands create definitions for a new target table in the SQL Server database with a remote schema of dbo and a name of TRGEMPLOYEE.

```
SET SERVER CONTROL TO DB SAMPLE;
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB MSSQLDB NONIBM SERVER SQLSERVER;
SET OUTPUT CAPTURE SCRIPT "cap.sql";
SET OUTPUT TARGET SCRIPT "target.sql";
SET OUTPUT CONTROL SCRIPT "control.sql";
SET LOG "MEM.OUT";
CREATE MEMBER IN SETNAME SET1 APPLYQUAL APPQUAL1 ACTIVATE YES
   SOURCE repldba.EMPLOYEE TARGET NAME repldba.TRGEMPNICK
   REMOTE SCHEMA dbo REMOTE TABLE TRGEMPLOYEE;
```
CREATE REGISTRATION command

Use the CREATE REGISTRATION command to register a source table, view, or nickname so that it can be used for replication. You can use this command to create multiple registrations using one command.

Syntax

```
CREATE REGISTRATION (objname, objowner) [RMTJRN LIB libname NAME journalname] [DIFFERENTIAL REFRESH diff-ref-clause] [FULL REFRESH ONLY diff-ref-clause]
```

diff-ref-clause:

```
STAGE cd_or_ccd_name [CONDENSED ON OFF] cd_or_ccd_owner
NONIBM fed-clause [IN DB name NAMING PREFIX prefix] prof-clause
COLS capcol-clause OPTIONS opt-clause
```

fed-clause:

```
remoteccdowner remoteccdname
```

prof-clause:

```
CREATE USING PROFILE pname [REUSE]
```

capcol-clause:

```
ALL IMAGE AFTER BOTH IMAGE AFTER BOTH (colname IMAGE AFTER BOTH) [PREFIX X befimgpref]
```

opt-clause:

```
CONFLICT NONE UPDATE AS DELETE INSERT OFF UPDATE AS DELETE INSERT ON CAPTURE ALL CHANGES
```

ASNCLP Program Reference for Replication and Event Publishing
Parameters

**objowner**
-Specifies the owner of the source object (table, view, or nickname) to register.
-You can specify multiple objects.

**objname**
-Specifies the name of the source object (table, view, or nickname) to register.
-You can specify multiple objects.

**LIB libname**
-Specifies the AS/400® library name.

**NAME journalname**
-Specifies the AS/400 journal name.

**DIFFERENTIAL REFRESH**
-Specify to update the target table periodically as the source object changes.

**FULL REFRESH ONLY**
-Specify to do a full refresh only, instead of applying changes.

**diff-ref-clause:**

**STAGE cd_or_ccd_owner.cd_or_ccd_name**
-Specifies the CD table owner and name. For non-DB2 sources, specifies the CCD table owner and name.

**Note:** If the object name is a view, then there can be multiple CD table names. Do not include this parameter because the command will generate view names for you. In this case, the ASNCLP program ignores any values you specify for this parameter.

**CONDENSED**

**ON**
-Specify to retain the most current data value.

**OFF**
-Specify to retain a history of data.

**Note:**
- Must be set to **OFF** if the source is non-DB2.
- This parameter is ignored for a CD table; CD tables are always noncondensed.

**NONIBM**
-Specifies the non-DB2 options.

**remoteccdowner**
-Specifies the CCD table owner in the non-DB2 database.

**remoteccdname**
-Specifies the CCD table name in the non-DB2 database.

**IN**
-Specifies the CD or CCD table space. If you do not specify the **IN** clause, the command uses the DB2 defaults for table spaces.
**DB name**
Specifies the name of an existing database where the CD or CCD table will be created. You must specify the database name, even if you set the database name in the profile.

**tsname**
Specifies the table space name. For z/OS, the name includes the database name (for example, "dbname.tsname"). You can specify a heterogeneous segment or table space name, but it must already exist.

**NAMING PREFIX prefix**
Specifies a naming prefix for the control tables.

**prof-clause:**

**CREATE USING PROFILE pname**
Specify to create the registration using a profile.

**REUSE**
Specify to reuse the current table space or index. You must issue the **CREATE USING PROFILE** parameter before you can use the **REUSE** parameter. When you specify the **REUSE** parameter, the ASNCLP program checks if the table space or index exists for the **tsname**:
- If the table space or index exists, the ASNCLP program resets the flags and passes the fully populated object to the API.
- If the table space or index does not exist, the ASNCLP program displays a syntax error saying that the **CREATE USING PROFILE** parameter is expected.

**COLS**
Specifies the columns that you want to register.

**Note:** This command only applies if the object is table. If the object is view, you can not have a registered sub set of columns.

**capcol-clause:**

**ALL**
Specifies that you want to register all columns. This is the default.

**IMAGE**

**AFTER**
Specify to register only after-image columns.

**BOTH**
Specify to register both after-image and before-image columns.

**colname**
Specifies a list of the columns that you want to register.

**PREFIX**
- If you specify **IMAGE AFTER**, the prefix will be null and the source will not allow any before-image columns.
- If you specify **IMAGE BOTH** or **IMAGE BEFORE** and do not specify **PREFIX**, a default value of X is used as a prefix for the before images. If you specify a **PREFIX**, that value is used.

You cannot alter an existing before-image prefix using the **ALTER REGISTRATION ROW** command. However, you can add that prefix to a new before-image column. If the existing before-image prefix is null and you want
to add a before-image column to the existing registration, you can specify the before-image prefix using the ALTER REGISTRATION ADD command. If you do not specify the prefix, the ASNCLP program sets it to a default value of X.

opt-clause:

CONFLICT
   Specifies the conflict-detection level.

   NONE
   No conflict detection. Conflicting updates between the master table and the replica table will not be detected. This option is not recommended for update-anywhere replication. This is the default.

   STANDARD
   Moderate conflict detection. During each Apply cycle, the Apply program compares the key values in the master’s CD table with those in the replica’s CD table. If the same key value exists in both CD tables, it is a conflict. In case of a conflict, the Apply program will undo the transaction that was previously committed at the replica by reading from the replica’s CD table and keeping only the changes that originated at the master.

   ENHANCED
   Conflict detection that provides the best data integrity among the master and its replicas. As with standard detection, the Apply program compares the key values in the master’s CD table with those in the replica’s CD table during each Apply cycle. If the same key value exists in both CD tables, it is a conflict. However, with enhanced detection, the Apply program waits for all inflight transactions to commit before checking for conflicts. To ensure that it catches all inflight transactions, the Apply program locks all target tables in the subscription set against further transactions and begins conflict detection after all changes are captured in the CD table. In case of a conflict, the Apply program will undo the transaction that was previously committed at the replica by reading from the replica’s CD table and keeping only the changes that originated at the master.

UPDATE AS DELETE INSERT

   ON
   Specify to capture updates as delete-insert pairs.

   OFF
   Specify to capture updates as updates. This is the default.

CAPTURE

   ALL
   Specify to capture everything. This is the default.

   CHANGES
   Specify to capture only changes.

FORWARDING

   OFF
   Specify not to forward changes from this source. This is the default.

   ON
   Specify to forward changes from this source.

FULL REFRESH
ON
Specify to allow full refreshes for this source. This is the default.

OFF
Specify not to allow full refreshes for this source.

STOP ON ERROR

ON
Specify not to stop the Capture program if it detects an error for this registration. This is the default.

OFF
Specify to stop the Capture program if it detects an error for this registration.

Usage notes

If multiple objects are registered at one time:

- The CD table or CCD table object owner and name clause is ignored; the command generates its own defaults.
- The table space specifications apply to all registrations.
- The OPTIONS values are common across all registrations.
- If the source object is view, the command decides whether the source can be registered as differential or full refresh and the user input will be ignored.

Example 1

To create a registration for DB2ADMIN.STAFF that only does full refreshes:

```
CREATE REGISTRATION (DB2ADMIN.STAFF) FULL REFRESH ONLY
```

Example 2

To create a registration for DB2ADMIN.STAFF that updates the target table as the source objects change, registers after-image columns C002 and C003, and registers both after-image and before-image columns C000 and C001:

```
CREATE REGISTRATION (DB2ADMIN.STAFF) DIFFERENTIAL REFRESH STAGE CDSTAFF
COLS (C000 IMAGE BOTH, C001 IMAGE BOTH, C002 IMAGE AFTER, C003 IMAGE AFTER) PREFIX X
```

CREATE STMT command

Use the CREATE STMT command to create a statement for an existing subscription set. This command lets you add a SQL statement or a stored procedure that Apply will process to the subscription set.

Syntax

```
CREATE STMT IN SETNAME setname APPLYQUAL applyqual
  SETTYPE R U P
  SQL "statement" NUMBER stmtnumber EXECUTE AT SOURCE
  PROC "procname" NUMBER stmtnumber
  AFTER AT TARGET
  BEFORE AT TARGET
```
Parameters

**SETNAME** setname
   Specifies the subscription-set name.

**APPLYQUAL** applyqual
   Specifies the Apply qualifier for the subscription set.

**SETTYPE**
   Specifies the subscription-set type.
   - **R**  Specifies a read-only set. This is the default.
   - **U**  Specifies an update-anywhere set.
   - **P**  Specifies a peer-to-peer set.

**SQL "statement"**
   Specifies an SQL statement. The double quotation marks are required.

**PROC "procnane"**
   Specifies a stored procedure name. The double quotation marks are required.

**NUMBER stmtnumber**
   Specifies the statement number to assign to this SQL statement or stored procedure. The default is (the value for the STMT_NUMBER column in the IBMSNAP_SUBS_STMT table) + 1.

**EXECUTE**
   Specifies where and when to execute the statement or procedure.
   
   **AT SOURCE**
      Specify to execute the statement or procedure at the source server.
   
   **AFTER AT TARGET**
      Specify to execute the statement or procedure at the target server after the Apply program processes the subscription set.
   
   **BEFORE AT TARGET**
      Specify to execute the statement or procedure at the target server before the Apply program processes the subscription set.

**SQLSTATES "states"**
   Specifies the SQL states that are accepted as normal during execution of the statement or procedure. The double quotation marks are required.

Example 1

To create a statement for the SET00 subscription set that executes an SQL statement at the source:
   CREATE_STMT IN SETNAME SET00 APPLYQUAL AQ00 SQL "statement" EXECUTE AT SOURCE

Example 2

To create a statement for the SET00 subscription set that executes the stored procedure at the target server before the Apply program processes the subscription set:
   CREATE_STMT IN SETNAME SET00 APPLYQUAL AQ00 PROC "procnane" EXECUTE BEFORE AT TARGET
CREATE SUBSCRIPTION SET command

Use the CREATE SUBSCRIPTION SET command to create an empty subscription set.

Syntax

```
CREATE SUBSCRIPTION SET SETNAME setname APPLYQUAL applyqual ACTIVATE NO YES ONCE
```

```
SETTYPE R U P
```

```
TIMING EVENT eventname INTERVAL minutes BOTH EVENT eventname INTERVAL minutes CONTINUOUS
```

```
START DATE "yyyy-mm-dd" TIME "hh:mm:ss.ffffff" NONIBM SOURCE SERVER srvname
```

```
COMMIT COUNT n
```

Parameters

**SETNAME setname**

Specifies the subscription-set name.

**APPLYQUAL applyqual**

Specifies the Apply qualifier for the subscription set.

**ACTIVATE**

Specifies whether to activate the subscription set.

**NO**

Specify to not activate the subscription set. This is the default.

**YES**

Specify to activate the subscription set.

**ONCE**

Specify to activate the subscription set for one Apply cycle, then deactivate the subscription set.

**SETTYPE**

Specifies the subscription-set type.

**R**

Specifies a read-only set. This is the default.

**U**

Specifies an update-anywhere set.

**P**

Specifies a peer-to-peer set.

**TIMING**

Specifies the timing for the subscription set.

**EVENT eventname**

Specifies the event that when posted to the IBMSNAP_SUBS_EVENT table, causes the Apply program to process the subscription set.
INTERVAL minutes
   Specifies the interval for the Apply program to process the subscription set. 
   The default interval is 20 minutes.

BOTH
   Specifies that this subscription set uses both event and interval timing.

CONTINUOUS
   Specifies that the Apply program should process the subscription set 
   continuously. This keyword is equivalent to specifying an interval of zero 
   minutes.

START DATE "yyyy-mm-dd"
   Specifies the date to activate the subscription set. The double quotation marks 
   are required.

TIME "hh:mm:ss.ffffff"
   Specifies the time to activate the subscription set. The double quotation marks 
   are required.

NONIBM SOURCE SERVER srvname
   Specifies the name of the non-DB2 source server.

COMMIT COUNT n
   Specifies the number of transactions that the Apply program should process 
   before issuing a SQL COMMIT statement for the subscription set. The default 
   value is NULL, which means that the Apply program issues just one COMMIT 
   statement for the subscription set after it processes the entire set. Do not 
   specify the COMMIT COUNT option if you want the default behavior.

Usage notes
   • This command can create only empty subscription sets, whereas the Replication 
     Center allows you to create empty subscription sets or add members to the set 
     while creating it.
   • A Capture schema is required, even though the set is empty.
   • Because the set is empty, the default for activating the set is NO.
   • To add a member to an existing subscription set, use the CREATE MEMBER 
     command.
   • To add a statement to the set, issue the CREATE SUBSCRIPTION SET STMTS 
     command.

Example 1

To create a subscription set SET00 that activates on 2006-11-22 at 09:00:00.000000:
CREATE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 ACTIVATE YES TIMING INTERVAL 1 
START DATE "2006-11-22" TIME "09:00:00.000000"

Example 2

To create a subscription set SET00 that activates for one Apply cycle on 2006-11-22 
at 09:00:00.000000:
CREATE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 ACTIVATE ONCE TIMING CONTINUOUS 
START DATE "2006-11-22" TIME "09:00:00.000000" NONIBM SOURCE SERVER SAMPLE
DROP CONTROL TABLES ON command

Use the DROP CONTROL TABLES ON command to drop a set of Capture, Apply, or Monitor control tables.

This command does not drop replication control tables on an OS/400® system.

Syntax

```
DROP CONTROL TABLES ON CAPTURE SERVER
  APPLY CONTROL SERVER ARCHLEVEL 0801
  MONITOR CONTROL SERVER ARCHLEVEL 0201 0805

NONIBM SCHEMA name
```

Parameters

CAPTURE SERVER
Specify to drop the Capture control tables.

APPLY CONTROL SERVER
Specify to drop the Apply control tables.

MONITOR CONTROL SERVER
Specify to drop the Monitor control tables.

ARCHLEVEL
Specifies the replication architecture level for the control tables that you want to drop.

- **0801**
  Specifies the Version 8 architecture level. For the Monitor control tables, the architecture level is always 0801.
  - **z/OS** 0801 specifies control tables created on a z/OS system running in version 8 compatibility mode.

- **0201**
  Specifies the architecture level for Version 5, Version 6, or Version 7.

- **0805**
  - **z/OS** Specifies the control tables created on a z/OS system running in new-function mode

NONIBM SCHEMA name
Specifies the remote schema name to use for heterogeneous replication. The following non-DB2 data sources are supported:

- Oracle
- Sybase
- Microsoft SQL Server
- Informix®
- Teradata

Usage notes
- The SET DROP command affects this command.
• This command drops the table spaces that the control tables are in if they do not contain any other objects.

• **Recommendation:** If the pre-Version 8 tables contain any data, migrate them instead of dropping them.

**Example 1**

To drop the Version 5 Capture control tables:

```
DROP CONTROL TABLES ON CAPTURE SERVER ARCHLEVEL 0201
```

**Example 2**

To drop the Version 8 Apply control tables:

```
DROP CONTROL TABLES ON APPLY CONTROL SERVER ARCHLEVEL 0801
```

**DROP MEMBER command**

Use the DROP MEMBER command to drop a member from an existing subscription set.

**Syntax**

```
DROP MEMBER FROM SETNAME setname APPLYQUAL applyqual SOURCE objowner..objname
```

```
TARGET objowner..objname
```

**Parameters**

**SETNAME setname**

Specifies the subscription-set name.

**APPLYQUAL applyqual**

Specifies the Apply qualifier for the subscription set.

**SOURCE objowner..objname**

Specifies the source object’s owner and name.

**TARGET objowner..objname**

Specifies the target object’s owner and name.

**Usage notes**

• For update-anywhere subscription sets, members for both replication directions (master-to-replica and replica-to-master) are dropped.

• The values specified in the SET DROP command determine whether the target table space is also dropped depends on the SET DROP command.

• Whether the target table is also dropped depends on the environment command:
  - If the target table has dependent subscription sets, it is not dropped and the autoregistration information is not deleted.
  - If there are no dependent subscription sets, the target table is dropped depending on the SET SERVER command. The autoregistration information is deleted.
Example

To drop a member from the SET00 subscription set:

```
DROP MEMBER FROM SETNAME SET00 APPLYQUAL AQ00 SOURCE DB2ADMIN.STAFF
TARGET DB2ADMIN.TRGSTAFF;
```

**DROP REGISTRATION command**

Use the DROP REGISTRATION command to drop one or more registrations.

**Syntax**

```
DROP REGISTRATION (objowner, objname)
```

**Parameters**

- `objowner`
  
  Specifies the owner of the source object (table, view, or nickname) to drop.

- `objname`
  
  Specifies the name of the source object (table, view, or nickname) to drop.

**Usage notes**

- The SET DROP command affects whether associated table spaces of the CD tables will be dropped when the objects are dropped.
- If the object is a view, only the CD views are dropped.
- For nicknames, this command does not drop the associated table spaces.

**Example 1**

To drop the registration for DB2ADMIN.STAFF:

```
DROP REGISTRATION (DB2ADMIN.STAFF)
```

**Example 2**

To drop the registration for DB2ADMIN.STAFF and DB2ADMIN.EMPLOYEE:

```
DROP REGISTRATION (DB2ADMIN.STAFF, DB2ADMIN.EMPLOYEE)
```

**DROP STMT command**

Use the DROP STMT command to drop SQL statements from an existing subscription set.

**Syntax**

```
DROP STMT FROM SETNAME setname APPLYQUAL applyqual
```

ASNCLP Program Reference for Replication and Event Publishing
Parameters

**SETNAME** *setname*
Specifies the subscription-set name.

**APPLYQUAL** *applyqual*
Specifies the Apply qualifier for the subscription set.

**SETTYPE**
Specifies the subscription-set type.
- **R** Specifies a read-only set. This is the default.
- **U** Specifies an update-anywhere set.
- **P** Specifies a peer-to-peer set.

**NUMBER** *stmtnumber*
Specifies the statement number to drop. You can specify multiple numbers using commas and parentheses.

Usage notes

- You cannot drop statements that are added to a subscription set for heterogeneous replication. These statements have the value G for the BEFORE OR AFTER column of the IBMSNAP_SUBS_STMTS table.

Example

To drop a statement from the subscription set SET00:

```
DROP STMT FROM SETNAME SET00 APPLYQUAL AQ00 NUMBER (5)
```
Example

To drop the subscription set SET00:
DROP SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00

**OFFLINE LOAD command**

Use the OFFLINE LOAD command to control a manual full refresh for offline load procedures.

You must first run the OFFLINE LOAD BEFORE command to prepare for an offline load. This will generate the scripts to deactivate the relevant subscription sets. After you have completed your offline load, you then need to run the OFFLINE LOAD AFTER command to reactivate the subscription set and reset the IBMSNAP_PRUNCNTL and IBMSNAP_SIGNAL tables.

**Syntax**

```
OFFLINE LOAD BEFORE SETNAME setname APPLYQUAL applyqual
```

**Parameters**

**BEFORE**
Specifies that you want to modify your replication environment in preparation for running an offline load for the target tables.

**AFTER**
Specifies that you want to modify your replication environment after running an offline load for the target tables.

**SETNAME** *setname*
Specifies the subscription-set name.

**APPLYQUAL** *applyqual*
Specifies the Apply qualifier for the subscription set.

**Example 1**

To run the OFFLINE LOAD BEFORE command and to generate the scripts to deactivate the subscription set SET00:
```
OFFLINE LOAD BEFORE SETNAME SET00 APPLYQUAL AQ00
```

**Example 2**

To run the OFFLINE LOAD AFTER command and to reactivate the subscription set SET00 and to reset the IBMSNAP_PRUNCNTL SET and IBMSNAP_SIGNAL tables:
```
OFFLINE LOAD AFTER SETNAME SET00 APPLYQUAL AQ00
```

**PROMOTE REGISTRATION command**

Use the PROMOTE REGISTRATION command to promote existing registrations.
Syntax

```plaintext
PREMITE REGISTRATION (objowner, objname) USING new-clause
```

table-clause:

```plaintext
CD SCHEMA cdschema CREATE SOURCE WITH SCHEMA tableschema
```

view-clause:

```plaintext
CD SCHEMA FOR VIEW viewschema SOURCE TABLE tableschema
```

```plaintext
CREATE SOURCE VIEW WITH UNREGISTERED BASE TABLES USING SCHEMA viewschema
```

Parameters

**objowner**

Specifies the owner of the source object (table, view, or nickname) to promote. You can specify multiple objects.

**objname**

Specifies the name of the source object (table, view, or nickname) to promote. You can specify multiple objects.

new-clause:

**SOURCE DB aliasname**

Specifies the new source database alias for the promoted object. This database is where you will run the generated script.

**CAPTURE SCHEMA schemaname**

Specifies the Capture schema to use when promoting a registration.

**TABLE**

Specifies a CD table.

**VIEW**

Specifies a CD view.

tbl-clause:

**CD SCHEMA cdschema**

Specifies the new CD-table schema name for the promoted object.
CREATE SOURCE WITH SCHEMA tableschema
   Specifies the new source-table schema name to use when promoting the underlying table.

view-clause:

CD SCHEMA FOR
   VIEW viewschema
      Specifies the new CD-view schema name for the promoted object.
   SOURCE TABLE tableschema
      Specifies the new CD-table schema name for the promoted object.

CREATE SOURCE VIEW
   Specify to promote the view on the new source.

WITH UNREGISTERED BASE TABLES
   Specify to promote underlying base tables that are not registered.

USING SCHEMA viewschema
   Specifies the new source-view schema name to use when promoting the underlying view and the unregistered base tables.

Usage notes

• If you do not specify the USING parameter, this command uses the existing values for the object.
• This command uses the following rules when generating the SQL scripts:
  – All views and tables referenced by the registered views exist on the new server.
  – All registered source tables referenced by the registered views are already promoted to the new server.
  – The WITH UNREGISTERED BASE TABLES clause promotes only the unregistered base tables of the view. It does not promote the registered base tables. You must promote the registered base tables separately before promoting the registered view.
  – The same new schema name will be used for both the underlying base tables and the view.
• The command does not support a new source CD schema when promoting subscription sets; do not change the CD schema when promoting registrations.

Example 1

To promote the registration for DB2ADMIN.STAFF using the SAMPLE database and ASN1 schema:
   PROMOTE REGISTRATION (DB2ADMIN.STAFF) USING SOURCE DB SAMPLE TABLE CD SCHEMA ASN1

Example 2

To promote the registration for DB2ADMIN.STAFF and to name the new CD-table schema STAFF:
   PROMOTE REGISTRATION (DB2ADMIN.STAFF) USING VIEW CD SCHEMA FOR SOURCE TABLE STAFF

PROMOTE SUBSCRIPTION SET command

   Use the PROMOTE SUBSCRIPTION SET command to recreate an existing subscription set in another replication environment.
Syntax

```
PROMOTE SUBSCRIPTION SET SETNAME setname APPLYQUAL applyqual
```

new-clause::

```
CAPTURE SCHEMA FOR SOURCE sourcename REPLICA replicaname
```

```
DB FOR SOURCE sourcealias TARGET targetalias CONTROL controlalias
```

```
APPLYQUAL newapplyqual SETNAME newsetname SOURCE SCHEMA newsourcename
```

```
TARGET SCHEMA newtargetname CD SCHEMA newcdschema
```

Parameters

**SETNAME setname**  
Specifies the subscription-set name.

**APPLYQUAL applyqual**  
Specifies the Apply qualifier for the subscription set.

**USING**  
Specifies the information for the promoted subscription set.

new-clause:

**CAPTURE SCHEMA FOR**  
Specifies the new Capture schema.

**SOURCE sourcename**  
Specifies the new Capture schema at the source.

**REPLICA replicaname**  
Specifies the new Capture schema at the source for a replica.

**DB FOR**  
Specifies the new database alias.

**SOURCE sourcealias**  
Specifies the new source database alias for the promoted object. This database is where you will run the generated script.

**TARGET targetalias**  
Specifies the new target database alias for the promoted object. This database is where you will run the generated script.

**CONTROL controlalias**  
Specifies the new Apply control database alias for the promoted object. This database is where you will run the generated script.

**APPLYQUAL newapplyqual**  
Specifies the new Apply qualifier.
SETNAME newsetname
   Specifies the new subscription-set name.

SOURCE SCHEMA newsourcename
   Specifies the new source schema name.

TARGET
   Specifies the schemas for the target.

SCHEMA newtargetname
   Specifies the new target schema name.

CD SCHEMA newcdschema
   Specifies the new target-CD schema name.

Usage notes
- If you do not specify a USING clause, this command uses the existing values.
- The command does not support a new source CD schema when promoting subscription sets, so you should not change the CD schema when you promote registrations.

Example
To promote an existing subscription set SET00:

```
PROMOTE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 USING CAPTURE SCHEMA
FOR SOURCE ASN2 SETNAME SET01 SOURCE SCHEMA SAMPLE1 TARGET SCHEMA TARGET1
CD SCHEMA ASN3
```

---

**SET CAPTURE SCHEMA command (SQL replication)**

Use the SET CAPTURE SCHEMA command to set a source and target Capture schema for all task commands. The default Capture schema is ASN. You can use this command to change the default.

This command allows you to omit the Capture schema settings in the task commands.

**Syntax**

```
SET CAPTURE SCHEMA SOURCE TO DEFAULT NULLS -capschema-
```

**Parameters**

**SOURCE**
   Specifies the Capture schema at the source. The schema can be any valid DB2 schema name.

**TARGET**
   Specifies the Capture schema at the target (used for autoregistration of replica or CCD target tables). The schema can be any valid DB2 schema name.

**DEFAULT**
   Specify to set the Capture schema to ASN and to reset any previous SET CAPTURE SCHEMA commands.

**NULLS**
   Specify to set the Q Capture schema to NULL.
capschema
   Specifies the name of a schema that generates the Capture control tables.

Example 1

To set the Capture schema to ASN by default:
SET CAPTURE SCHEMA SOURCE TO DEFAULT

Example 2

To set the Capture schema to ASN1:
SET CAPTURE SCHEMA SOURCE ASN1

SET DROP command (SQL replication)

Use the SET DROP command to determine whether to drop the table space when you drop the database object (replication control tables, registrations, or subscription-set members).

Note: The drop options affect multiple objects (that is, they are at the environment-command level), whereas the create options are at an object level (that is, they are at the task-command level).

Syntax

```
SET DROP TARGET ALWAYS
     NEVER
```

```
SET DROP CD TABLESPACE WHEN EMPTY
       CCD CCD
       TARGET CONTROL TABLES
     NEVER
```

Parameters

TARGET
   Specifies whether you want to drop the target tables with the subscription.

   ALWAYS
      Always drop the target table.

   NEVER
      Never drop the target table.

DROP
   Specifies what you want to drop with the subscription.

   CD
      Change data table

   CCD
      Consistent-change-data table

   TARGET
      Target table

   CONTROL TABLES
      Capture, Apply, or Monitor control tables
These options are relevant only for operating-system environments for which the commands create the table spaces. You can always specify the drop flag for each of these object types.

**TABLESPACE**
Specifies when to drop the table space that contains the specified object.

**WHEN EMPTY**
Drop the table space only when it is empty.

**NEVER**
Never drop the table space.

**Usage notes**

The drop subscription-set member command decides whether to drop an autoregistered target table. If the autoregistration has dependent subscriptions, the command does not drop the target table and does not drop the registration; otherwise, the registration and the target table are dropped only if the SET DROP TARGET ALWAYS command allows it.

**Example 1**

To always drop the target table’s table space when the subscription is dropped:

```
SET DROP TARGET ALWAYS
```

**Example 2**

To drop the CCD table space when it is empty:

```
SET DROP CCD TABLESPACE WHEN EMPTY
```

---

**SET LOG command**

Use the SET LOG command to define the log file for the ASNCLP session. The log file contains informational warning and error messages.

**Syntax**

```
SET LOG "logfilename"
```

**Parameters**

"logfilename"
Specifies the output log file name. The default file name is replmsg.log.

**Usage notes**

- If the files already exist, the ASNCLP program will append to them.
- The double quotation marks in the command syntax are required.

**Example**

To name the output log file cnsrc.err:

```
SET LOG "cnsrc.err"
```
SET OUTPUT command (SQL replication)

Use the SET OUTPUT command to define output files for the ASNCLP session. The output files contain the SQL statements needed to set up replication.

Syntax

```plaintext
SET OUTPUT CAPTURE SCRIPT "capfname" CONTROL SCRIPT "cntlfname"
TARGET SCRIPT "trgfname" MONITOR SCRIPT "monfname"
```

Parameters

- **CAPTURE SCRIPT "capfname"**
  Specifies the output file name for SQL scripts that run at the Capture server. The default file name is replcap.sql.

- **CONTROL SCRIPT "cntlfname"**
  Specifies the output file name for SQL scripts that run at the Apply control server. The default file name is replctl.sql.

- **TARGET SCRIPT "trgfname"**
  Specifies the output file name for SQL scripts that run at the target server. The default file name is repltrg.sql.

- **MONITOR SCRIPT "monfname"**
  Specifies the output file name for scripts that run at the Monitor control server. The default file name is replmonitor.sql.

Usage notes

- If you do not need an output file, run the SET OUTPUT command and specify "" for the file name.
- If a script already exists, the new script appends to the current script.
- The double quotation marks in the command syntax are required.

Example 1

To name the output Apply control script file control.sql:
```
SET OUTPUT CONTROL SCRIPT "control.sql"
```

Example 2

To name the output monitor script file monitor.sql:
```
SET OUTPUT MONITOR SCRIPT "monitor.sql"
```

SET PROFILE command (SQL replication)

Use the SET PROFILE command to customize rules for creating table space objects. After you issue a SET PROFILE command, all subsequent task commands inherit the table space DDL specifications defined by the command. You can associate a profile with a task command by specifying the profile's name in the task command.
Syntax

```
SET PROFILE profilename prof-clause
```

prof-clause:

```
FOR OBJECT CD CCD TARGET UOW OTHERS PAGE LOCK ROW LOCK TABLESPACE OPTIONS zos-tbs-clause uw-tbs-clause
```

zos-tbs-clause:

```
ZOS OR db-name BUFFERPOOL bufferpoolname ENCODING EBCDIC ASCII UNICODE

STOGROUP stogroupname PRIQTY ABSOLUTE PERCENT OF SOURCE SECQTY ABSOLUTE PERCENT OF SOURCE
```

uw-tbs-clause:

```
UW BUFFERPOOL bufferpoolname PAGESIZE n

USING FILE "container" SIZE n PAGES

DEVICE KILO MEGA GIGA PERCENT OF SOURCE n
```

Parameters

**PROFILE profilename**
Specifies the profile name.

**UNDO**
Specify to undo a specific profile.

prof-clause:

**FOR OBJECT**
Specify to set an object for the table space options:

**CD**
Change data table

**CCD**
Consistent change data table
TARGET
Target table

UOW
Unit-of-work table

OTHERS
All other control tables, except the UOW table

PAGE LOCK
z/OS: All tables that follow the page locking mechanism

ROW LOCK
z/OS: All tables that follow the row locking mechanism

TABLESPACE OPTIONS
Specify to set the table space options. You can specify table space options for z/OS or Linux, UNIX, and Windows.

z/OS: No support for table space lock size because the replication API infers the correct value in most cases.

Linux, UNIX, and Windows:
• The ASNCLP program supplies the MANAGED BY DATABASE clause.
• No support for LARGE table spaces.
• No support for heterogeneous replication environments.

zos-tbs-clause:

DB dbname
Specifies the name of the z/OS database to connect to. This parameter does not specify the subsystem name; use the SET SERVER command to set the subsystem name to connect to.

BUFFERPOOL bufferpoolname
Specifies the buffer pool name.

ENCODING
Specifies the encoding scheme (EBCDIC, ASCII, or UNICODE). The default is EBCDIC.

STOGROUP stogroupname
Specifies a storage group name.

PRIQTY
Specify to set the minimum primary space allocation for a DB2-managed data set for a table space.

SECQTY
Specify to set the minimum secondary space allocation for a DB2-managed data set for a table space.

ABSOLUTE
Specifies an actual value in kilobytes (denoted as \texttt{n} or \texttt{m} in the syntax diagram) for space allocation. See the information about the CREATE TABLESPACE command for more details.

PERCENT OF SOURCE
Specifies the percentage of the source table size, as indicated by:
• z/OS The column “npages” in SYSIBM.SYSTABLES
• Linux UNIX Windows The column “npages” in SYSTAT.TABLES
This method will work only if the column holds the correct value for this table, which can be achieved by running the “db2 runstats on table a.b.” command or by manually updating the DB2 catalog.

uw-tbs-clause:

**BUFFERPOOL** bufferpoolname

Specifies the buffer pool name.

**PAGESIZE** n

Specifies the page size of the table space.

**FILE**

Specifies the container path string for the File. For example, for UNIX you can set the container path to `/tmp/db/ts/` and for Windows, you can set the container path to `D:\tmp\db\ts\`.

**DEVICE**

Specifies the container path string for the device. For example, for UNIX you can set the container path to `/tmp/db/ts/` and for Windows, you can set the container path to `D:\tmp\db\ts\`.

"**container**"

Specifies the name of the container. The ASNCLP program will generate and append the table space name to the specified path when you run a task command such as CREATE REGISTRATION. The double quotation marks in the syntax are mandatory.

**SIZE** n

Specifies the size of the container:

**PAGES**

Actual number of pages

**KILO**

Kilobytes

**MEGA**

Megabytes

**GIGA**

Gigabytes

**Usage notes**

- You cannot specify your own naming convention for CD table names or table spaces because the task commands generate default values.
- This command is not used for heterogeneous replication environments because the task commands do not create table spaces on remote servers.
- OS/400 systems do not have table spaces that require special DDL.
- The task commands allow you to specify a table space clause so that you can use an existing table space. The task commands do not provide an index clause because indexes are always created (except in certain cases when creating target tables).
- The scope of the profile lasts only as long as the current session. Once you quit the ASNCLP session, the profile information is not saved for the next session.
Example 1

To create a profile TBSPROFILE that sets the table space options for the target control tables:

SET PROFILE TBSPROFILE FOR OBJECT TARGET TABLESPACE OPTIONS UW
USING FILE "c:\TSTRG.TS" SIZE 700 PAGES

Example 2

To undo the profile TBSPROFILE:

SET PROFILE TBSPROFILE UNDO

SET RUN SCRIPT command (SQL replication)

Use the SET RUN SCRIPT command to control whether to automatically run SQL statements that are generated by each ASNCLP task command before processing the next command or to manually run them later in a DB2 command prompt.

"Using SET RUN SCRIPT options" on page 52 helps you understand when to run commands immediately and when to run them later.

Syntax

```
SET RUN SCRIPT LATER
NOW STOP ON SQL ERROR ON OFF
```

Parameters

**LATER**
Specify to run the SQL scripts at a later time. If you specify to run them later, you must run the generated SQL script manually at a DB2 command prompt by using the following command:

db2 -tvf filename

where *filename* is the name of the SQL script file.

**Federated sources**: Use the following command to run the script for federated (non-DB2) sources:

db2 -td# -vf filename

**NOW**
Specify to automatically execute the SQL scripts.

**STOP ON SQL ERROR**
Specifies whether to stop running the SQL scripts if an error occurs.

**ON**
Specify to stop processing the ASNCLP commands when the first SQL statement fails. All previous SQL statements related to this command will be rolled back. If the source scripts run correctly and have been committed, and the target scripts have an error, only the target scripts will be rolled back. The committed source statements will not be rolled back.

**OFF**
Specify to process the ASNCLP commands and run all of the SQL statements, regardless of errors.
Using SET RUN SCRIPT options

Some ASNCLP CREATE commands require that one or more replication objects exist before the command can be processed. For example, you cannot create subscriptions until control tables exist.

These dependencies can influence whether you use the NOW or LATER options. In general, the following guidelines apply:

- If you want to create different types of objects in a single ASNCLP script, you are likely to need to use SET RUN SCRIPT NOW.
- If you have multiple ASNCLP scripts, each creating one or more instances of an object, you can use either NOW or LATER. If you use LATER, you are likely to need to run the generated SQL from one ASNCLP script before processing subsequent ASNCLP scripts.
- In some situations, objects of the same type require that SET RUN NOW be used.

[Figure 1 on page 53] shows these dependencies for SQL replication.
Usage notes

- Use SET RUN SCRIPT LATER when you want to verify the SQL scripts before you run them to create or update your replication configuration.
- Use SET RUN SCRIPT LATER if you want to create SQL script files on one operating system, but run them on another.
- This command supports scripts to set up heterogeneous replication. Federated registration generates a script that creates a trigger on the IBMSNAP_PRUNCNTL table to prune from all CCD tables. This trigger is dropped and recreated for each registration by including all of the previous registration information along with the current registration. If each registration script is not executed before the next registration script is run, the prune control trigger in the database does not have the CCD information for the previous
registration, and the trigger will be out of sync with the actual registered objects in the database. This problem can be solved by using the SET RUN SCRIPT NOW option for the input file.

Example 1

To run the SQL scripts at a later time:

SET RUN SCRIPT LATER

Example 2

To automatically run the SQL scripts but stop processing the ASNCLP commands if an error occurs:

SET RUN SCRIPT NOW STOP ON SQL ERROR ON

SET SERVER command (SQL replication)

Use the SET SERVER command to specify the remote iSeries™ source server, Capture control server, Apply control server, or target server to use in the ASNCLP session. After you set a server name, all subsequent commands in the session will apply to this server until you change the server with this command.

The SET SERVER command is required for the following task commands:

All control table commands
Set the Capture control server or Apply control server before creating or dropping replication control tables.

All registration commands (including promote)
Set the Capture control server before running the registration commands. For iSeries, you must also set the remote source server.

All subscription commands (including promote)
Set the Capture control, Apply control, and target servers before running the subscription commands, unless one or more servers are not needed. For example, because the ALTER SUBSCRIPTION SET and ALTER SUBSCRIPTION SET MEMBER commands modify only control tables on the Apply control server, you do not need to set the Capture control servers for these commands. For iSeries, you must set the remote source server.

Syntax

```
SET SERVER ALL
  -REMOTE SOURCE-
  -CAPTURE-
  -CONTROL-
  -TARGET-

TO NULLS
  -DB dbalias-
  -DBALIAS aliasname-
  -DBNAME dbname-
  -NONIBM SERVER remsrvr-

other-options
```
other-options:

- AS400 HOSTNAME — "hostname"
- ID — userid
- PASSWORD — pwd

Parameters

ALL
Specify to set the database for all servers (remote source server, Capture control server, Apply control server, target server).

REMOTE SOURCE
Specify to set the database as a remote source server.

CAPTURE
Specify to set the database as a Capture control server.

CONTROL
Specify to set the database as an Apply control server.

TARGET
Specify to set the database as a target server.

NULLS
Specify to set the server name to NULL. This option resets a previously set server name.

DB dbalias
Specifies the database alias name.

DBALIAS aliasname
Specifies the database alias name.

DBNAME dbname
Specifies the database name.

NONIBM SERVER remsrv
Capture control servers and target servers only: Specifies the remote server name for a non-DB2 source or target. This parameter is valid only for Capture control servers and target servers, not for Apply control servers.

AS400 HOSTNAME "hostname"
Specifies the OS/400 host name, typically an IP address or name.

ID userid
Specifies the user ID to use to connect to the database.

PASSWORD pwd
Specifies the password to use to connect to the database. If you specify the user ID and do not specify the password, you will be prompted to enter the password.

Usage notes
- Use the NONIBM SERVER clause to set up replication using non-DB2 data sources and targets such as Oracle and Sybase. The environment command saves the database server information, but does not perform the actual db2 connect command. The environment command assigns a database alias to a logical
replication server. The ASNCLP program attempts the connection to determine the platform and build the appropriate objects for the task commands.

- If you issue multiple environment commands, the most recent command overrides the current settings for a given remote source server, Capture control server, Apply control server, or target server. That is, you can associate only one value for each of these servers, but these values need not be the same.

**Example 1**

To set all servers to the database SAMPLE:

```
SET SERVER ALL TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd"
```

**Example 2**

To set the Capture control server to the database SAMPLE:

```
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd"
```

**Example 3**

To set the Capture control server and specify only the user ID in the command:

```
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN
```

You are prompted to enter the password. If you are running the commands from an input file in batch mode, the program waits for you to enter the password before the program processes the next commands.

**SET TRACE command**

Use the SET TRACE command to enable and disable the internal trace for the ASNCLP commands.

**Syntax**

```
SET TRACE [OFF | ON]
```

**Parameters**

- **OFF**
  
  Specify to turn off the trace.

- **ON**
  
  Specify to turn on the trace.

**Usage notes**

- The trace is written to stdout and stderr.

**Example 1**

To turn off the internal trace for the ASNCLP program:

```
SET TRACE OFF
```
Chapter 3. ASNCLP commands for Q replication and Classic replication

The ASNCLP commands for Q replication are divided into commands for unidirectional replication and commands for multidirectional replication.

The commands that are used for Classic replication are the same as the commands for unidirectional Q replication from relational sources (Classic replication is a type of unidirectional Q replication). Some of the parameters within these commands are unique to Classic replication.

Sample ASNCLP scripts for Q replication

The following sample scripts show you how to put together ASNCLP commands to set up unidirectional, bidirectional, and peer-to-peer Q replication.

Sample ASNCLP scripts for setting up unidirectional Q replication

This sample contains four ASNCLP scripts for setting up a unidirectional Q replication environment. It includes Q Capture and Q Apply control tables, a replication queue map, and a Q subscription.

ASNCLP scripts typically generate one or more SQL scripts to create replication objects. Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Q Capture and Q Apply control tables
2. Replication queue map
3. Q subscription

The final ASNCLP script checks the WebSphere MQ environment for Q replication and does not generate SQL statements. You must create the control tables and queue map before you run this script.

This sample has a section for each ASNCLP script, which you can copy to a text file and run by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

Table 3 on page 60 below the sample describes the SQL scripts that are generated.

For help creating the WebSphere MQ objects that are used in these scripts, see Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing and WebSphere MQ setup scripts for Q replication

ASNCLP script 1 (Q Capture and Q Apply control tables)

This script generates SQL statements that create Q Capture control tables at the SAMPLE database and Q Apply control tables at the TARGET database. It includes commands for the following tasks:
Setting the environment

Creating Q Capture control tables

Creating Q Apply control tables

Ending the ASNCLP session

---

1. Setting the environment
2. Creating Q Capture control tables
3. Creating Q Apply control tables
4. Ending the ASNCLP session

---

ASNCLP script 2 (replication queue map)

This script generates SQL statements to create a replication queue map. It includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map
3. Ending the ASNCLP session

---

ASNCLP SESSION SET TO Q REPLICAION;
SET LOG "rmap.err";
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
SET CAPTURE SCHEMA SOURCE ASN1;
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "QM2" FOR APPLY SCHEMA;
SET APPLY SCHEMA ASN1;
SET OUTPUT CAPTURE SCRIPT "qcapctrl.sql" TARGET SCRIPT "qappctrl.sql";
SET RUN SCRIPT LATER;
QUIT;

---

ASNCLP Program Reference for Replication and Event Publishing

58
SET CAPTURE SCHEMA SOURCE ASN1;
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA ASN1;
SET OUTPUT CAPTURE SCRIPT "qcapsub.sql" TARGET SCRIPT "qappmap.sql";
SET RUN SCRIPT LATER;

# 2 Creating a replication queue map.
# This command generates SQL to create a replication queue map,
# SAMPLE ASN1 TO TARGET ASN1. It specifies a remote administration
# queue and receive queue at the Q Apply server, and a send queue at
# the Q Capture server. The command also sets the number of agent threads
# for the Q Apply program to 8 (half of the default 16), and specifies that
# heartbeat messages be sent every 5 seconds.

CREATE REPLMAP SAMPLE ASN1 TO TARGET ASN1 USING
ADMINQ "ASN1.QM1.ADMINQ" RECVQ "ASN1.QM1_TO_QM2.DATAO"
SENDQ "ASN1.QM1_TO_QM2.DATAO" NUM APPLY AGENTS 8 HEARTBEAT INTERVAL 5;

# 3 Ending the ASNCLP session.
QUIT;

ASNCLP script 3 (Q subscription)

This script generates SQL statements to create a Q subscription. It specifies a
source table, EMPLOYEE, at the SAMPLE database, and a new target table,
TGTEMPLOYEE, at the TARGET database. The script includes commands for the
following tasks:

1. Setting the environment
2. Creating a Q subscription
3. Ending the ASNCLP session

# 1 Setting the environment.
# The SET OUTPUT command creates two SQL scripts: qcapsub.sql, which adds
# definitions for the Q subscription to the Q Capture control tables, and
# qappsub.sql, which adds definitions for the Q subscription to the Q Apply
# control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "qsub.err";
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE ASN1;
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA ASN1;
SET OUTPUT CAPTURE SCRIPT "qcapsub.sql" TARGET SCRIPT "qappsub.sql";
SET RUN SCRIPT LATER;

# 2 Creating the Q subscription
# This command generates SQL to create a Q subscription named EMPLOYEE0001
# that specifies the EMPLOYEE table as a source. The TARGET NAME keywords
# are used without the EXISTS or NAMING PREFIX keywords, resulting in a target
# table name of TGTEMPLOYEE. The EMPNO column, which is the primary key for the
# EMPLOYEE table, is specified as the key for replication. The command also
# specifies that the Q Apply program load the target table (LOAD PHASE 1) using
# the EXPORT and IMPORT utilities (LOAD TYPE 2).

CREATE QSUB USING REPLMAP SAMPLE ASN1 TO TARGET ASN1
(SUBNAME EMPLOYEE0001 EMPLOYEE OPTIONS HAS LOAD PHASE 1
TARGET NAME EMPLOYEE KEYS (EMPNO) LOAD TYPE 2);

# 3 Ending the ASNCLP session.
QUIT;
ASNCLP script 4 (check WebSphere MQ environment)

This script does not generate SQL. Instead, it checks whether the queue managers and queues that were specified in the other scripts exist, and whether the objects have the correct properties for Q replication. Then it checks the message flow between the queues in the replication queue map by sending test messages. The script includes commands for the following tasks:

1. Setting the environment
2. Checking the queue managers and queues
3. Sending test messages
4. Ending the ASNCLP session

# 1 Setting the environment.
# No SET RUN statement is required. The commands run immediately and send # results to the command window and log.

ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "qchecks.err";
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
SET CAPTURE SCHEMA SOURCE ASN1;
SET SERVER TARGET TO DB2ALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "QM2" FOR APPLY SCHEMA;
SET APPLY SCHEMA ASN1;

# 2 Checking the queue managers and queues.
# These commands check whether the queue managers and queues exist, and validate # their settings against the requirements for Q replication. If errors are detected, # you must correct them before you start the Q Capture and Q Apply programs.

VALIDATE WSMQ ENVIRONMENT FOR CAPTURE SCHEMA;
VALIDATE WSMQ ENVIRONMENT FOR APPLY SCHEMA;
VALIDATE WSMQ ENVIRONMENT FOR REPLQMAP SAMPLE_ASN1_TO_TARGET_ASN1;

# 3 Sending test messages.
# This command puts a test message on the send queue, ASN1.QM1_TO_QM2.DATAQ,
# and tries to get the message from the receive queue, ASN1.QM1_TO_QM2.DATAQ.
# The command also puts a test message on the Q Apply administration queue,
# ASN1.QM1.ADMINQ, and tries to get the message from the Q Capture administration
# queue, ASN1.QM1.ADMINQ.

VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP SAMPLE_ASN1_TO_TARGET_ASN1;

# 4 Ending the ASNCLP session.
QUIT;

Output of the scripts

In addition to the log files, this example produces six SQL script files in the same directory where you run the ASNCLP program. Table 3 describes the files.

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>qcapctrl.sql</td>
<td>Create Q Capture control tables</td>
</tr>
<tr>
<td>qappctrl.sql</td>
<td>Create Q Apply control tables</td>
</tr>
<tr>
<td>qcapqmap.sql</td>
<td>Insert definitions for a replication queue map into the Q Capture control tables</td>
</tr>
<tr>
<td>qappqmap.sql</td>
<td>Insert definitions for a replication queue map into the Q Apply control tables</td>
</tr>
</tbody>
</table>
Sample ASNCLP scripts for setting up unidirectional Q replication from a Classic data source

This sample contains three ASNCLP scripts for setting up a unidirectional Q replication environment from a Classic data source. It includes Q Apply control tables, a replication queue map, and a Q subscription.

ASNCLP scripts typically generate one or more SQL scripts to create replication objects. Table 4 on page 63 describes the SQL scripts that you create by running the samples. To create a Q subscription for a Classic source:
1. Use Classic Data Architect to create a relational mapping of the source table on the Classic server.
2. Create a Classic replication configuration file.
3. Create the Q Apply control tables
4. Update the capture parameters for the Classic data source
5. Create the replication queue map
6. Create the Q subscription

This sample has a section for each ASNCLP script, which you can copy to a text file and run by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

For help creating the WebSphere MQ objects that are used in these scripts, see Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing and WebSphere MQ setup scripts for Q replication

ASNCLP script 1: Create Q Apply control tables and update the capture parameters for the Classic data source

This script generates SQL statements that create Q Apply control tables at the TARGET database. The script includes commands for the following tasks:

1. Setting the environment
2. Creating Q Apply control tables
3. Update the capture parameters for the Classic data source
4. Ending the ASNCLP session

# 1 Setting the environment
# In the SET SERVER command, the user ID and password are optional. If you omit these keywords, the ASNCLP will use the implicit ID and password for connecting # to the database.
# The SET LOG command directs ASNCLP messages to the log file qcontrol.err.
# The SET OUTPUT command creates the classicctrl.sql SQL script, which creates Q Apply control tables at the TARGET database.
# The SETQMANAGER commands are required for creating Q Replication control tables.

ASNCLP SESSION SET TO Q REPLICATION;

Table 3. SQL script files that are created by the sample ASNCLP scripts (continued)

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>qcapqsub.sql</td>
<td>Insert definitions for a Q subscription into the Q Capture control tables</td>
</tr>
<tr>
<td>qappqsub.sql</td>
<td>Insert definitions for a Q subscription into the Q Apply control tables</td>
</tr>
</tbody>
</table>
SET LOG "qcontrol.err";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET QMGR "QM2" FOR APPLY SCHEMA;
SET APPLY SCHEMA ASN1;
SET OUTPUT TARGET SCRIPT "classicctrl.sql";

# 2 Creating Q Apply control tables
# This command specifies a password file, asnwd.aut. The Q Apply program uses this
# file to connect to the Classic data source when it loads the target table.

CREATE CONTROL TABLES FOR APPLY SERVER IN UW TBSPACE TSQAPP;
# 3 Update the capture parameters for the Classic data source
# The following commands update the IBMQREP_CAPPARMS table to add parameters
# that specify the WebSphere MQ queue manager and queues that are used by
# the Classic capture components.

SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE "asnservers.ini"
ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT NOW;
ALTER CAPTURE PARAMETERS QMGR asnqmgr RESTARTQ asnrestart ADMINQ asadmin;

# 4 Ending the ASNCLP session
QUIT;

**ASNCLP script 2: Create the replication queue map**

This script generates SQL statements to create a replication queue map. The script includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT command creates the qappmap.sql SQL script,
# which adds definitions for the queue map to the Q Apply
# control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "rmqmap.err";
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE "asnservers.ini"
ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA ASN1;
SET OUTPUT TARGET SCRIPT "qappmap.sql";

# 2 Creating a replication queue map
# This command generates SQL to create a replication queue map,
# CLASSIC ASN1 TO TARGET ASN1. It specifies a remote administration
# queue and receive queue at the Q Apply server, and a send queue at
# the Q Capture server. The command also sets the number of agent threads
# for the Q Apply program to 8 (half of the default 16), and specifies that
# heartbeat messages be sent every 5 seconds.

CREATE REPLQMAP CLASSIC ASN1 TO TARGET ASN1 USING
ADMINQ "ASN1.QM1.ADMINQ" RECVQ "ASN1.QM1_TO_QM2.DATAQ"
SENDQ "ASN1.QM1_TO_QM2.DATAQ" NUM APPLY AGENTS 8 HEARTBEAT INTERVAL 5;

# 3 Ending the ASNCLP session
QUIT;
ASNCLP script 3: Create the Q subscription

This script generates SQL statements to create a Q subscription. It specifies a source table, EMPLOYEE which is mapped to the Classic source through Classic Data Architect, and a new target table, TGTEMPLOYEE. The script includes commands for the following tasks:

1. Setting the environment
2. Creating a Q subscription
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT command creates the qappsub.sql SQL script,
# which adds definitions for the Q subscription to the Q Apply
# control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "qsub.err";
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE "asnservers.ini"
ID CLASSICADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET APPLY SCHEMA ASN1;
SET OUTPUT TARGET SCRIPT "qappsub.sql";

# 2 Creating the Q subscription
# This command generates SQL to create a Q subscription named CLASSIC0001
# that specifies the CLASSICTABLE table as a source. The TARGET NAME keywords
# are used without the EXISTS or NAMING PREFIX keywords, resulting in a target
# table name of TGTCLASSICTABLE. The command also specifies that the Q
# Apply program load the target table (LOAD PHASE I) using LOAD TYPE 4.

CREATE QSUB USING REPLQMAP CLASSIC ASN TO TARGET ASN1
(SUBNAME CLASSIC0001 CLASSICTABLE OPTIONS HAS LOAD PHASE I
TARGET NAME CLASSICTABLE LOAD TYPE 4);

# 3 Ending the ASNCLP session

QUIT;

Output of the scripts

Table 4 describes the SQL scripts that the ASNCLP sample scripts create.

<table>
<thead>
<tr>
<th>Output file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>classicctrl.sql</td>
<td>Creates Q Apply control tables</td>
</tr>
<tr>
<td>qappqmap.sql</td>
<td>Inserts definitions for a replication queue map into the Q Apply control tables</td>
</tr>
<tr>
<td>qappqsub.sql</td>
<td>Inserts definitions for a Q subscription into the Q Apply control tables</td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up bidirectional Q replication

This sample contains six ASNCLP scripts for setting up a bidirectional Q replication environment. It includes Q Capture and Q Apply control tables at both servers, replication queue maps in both directions, and two bidirectional Q subscriptions.
The scenario is a standby configuration with two databases, SAMPLE (the primary server) and SAMPLE2 (the standby server). One table, EMPLOYEE, will be replicated in both directions between the two databases. The Q Capture and Q Apply programs at the SAMPLE database have the schema RED. The two corresponding programs at the SAMPLE2 database have the schema BLUE.

The ASNCLP scripts create eight SQL scripts. Table 5 on page 68 below the sample describes each SQL script.

Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Q Capture and Q Apply control tables at SAMPLE
2. Q Capture and Q Apply control tables at SAMPLE2
3. Replication queue map from SAMPLE to SAMPLE2
4. Replication queue map from SAMPLE2 to SAMPLE
5. Q subscriptions

You can copy the commands for each ASNCLP script to a text file, modify the values, and run the script by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

For help creating the WebSphere MQ objects that are used in these scripts, see Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing and WebSphere MQ setup scripts for Q replication.

**ASNCLP script 1**

This script creates control tables at the SAMPLE database. It includes commands for the following tasks:
1. Setting the environment
2. Creating Q Capture and Q Apply control tables at the SAMPLE database
3. Ending the ASNCLP session

```
# 1  Setting the environment
# The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLE.sql, that is
# automatically named after the database. The script contains SQL statements to
# create both Q Capture and Q Apply control tables.
# The SET LOG command directs ASNCLP messages to one log file, bidir1.err.
# The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
# they are run.

ASNCLP SESSION SET TO Q REPPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "bidir1.err";
SET RUN SCRIPT LATER;

# 2  Creating Q Capture and Q Apply control tables at SAMPLE
# To use the script, change the ID and PASSWORD values.
# Both the Q Capture and Q Apply control tables will have the schema RED.

SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE RED;
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "RED.QM1.RESTARTQ" ADMINQ "RED.QM1.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA RED;
SET QMANAGER "QM1" FOR APPLY SCHEMA;
```
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# 3 Ending the ASNCLP session
QUIT;

**ASNCLP script 2**

This script creates control tables at the SAMPLE2 database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at the SAMPLE2 database
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLE2.sql. The script contains SQL statements to create both Q Capture and Q Apply control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "bidir2.err";
SET RUN SCRIPT LATER;

# 2 Creating Q Capture and Q Apply control tables at SAMPLE2
# Both the Q Capture and Q Apply control tables will have the schema BLUE.

SET SERVER CAPTURE TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE BLUE;
SET QMANAGER "QM2" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "BLUE.QM2.RESTARTQ" ADMINQ "BLUE.QM2.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA BLUE;
SET QMANAGER "QM2" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# 3 End the ASNCLP session.
QUIT;

**ASNCLP script 3**

This script creates a replication queue map from SAMPLE to SAMPLE2. It includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map from SAMPLE to SAMPLE2
3. Ending the ASNCLP session

# 1 Setting the environment
# Two SET SERVER commands are required because the replication queue map from SAMPLE to SAMPLE2 is defined in the Q Capture control tables at SAMPLE and the Q Apply control tables at SAMPLE2.
# The SET OUTPUT command specifies two SQL scripts: rqmred1.sql, which adds definitions to SAMPLE, and rqmbue1sql, which adds definitions to SAMPLE2.

ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE RED;
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA BLUE;
SET OUTPUT CAPTURE SCRIPT "rqmred1.sql" TARGET SCRIPT "rqmbue1.sql";
SET LOG "bidir3.err";
SET RUN SCRIPT LATER;
# 2 Creating a replication queue map
The CREATE REPLQMAP command specifies an administration queue and receive queue within the queue manager QM2 that is used for SAMPLE2, and a send queue within the queue manager QM1 that is used for SAMPLE.

CREATE REPLQMAP SAMPLE.RED_TO_SAMPLE2.BLUE USING
ADMINQ "BLUE.QM1.ADMINQ" RECVQ "BLUE.QM1_TO_QM2.DATAQ"
SENDQ "RED.QM1_TO_QM2.DATAQ";

# 3 Ending the ASNCLP session
QUIT;

### ASNCLP script 4

This script creates a replication queue map from SAMPLE2 to SAMPLE. It includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map from SAMPLE2 to SAMPLE
3. Ending the ASNCLP session

# 1 Setting the environment
Two SET SERVER commands are required because the replication queue map from SAMPLE2 to SAMPLE is defined in the Q Capture control tables at SAMPLE2 and the Q Apply control tables at SAMPLE.

# 2 Creating a replication queue map
The SET OUTPUT command specifies two SQL scripts, rqmblue2.sql, which adds definitions to SAMPLE2, and rqmred2.sql, which adds definitions to SAMPLE.

ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE BLUE;
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA RED;
SET OUTPUT CAPTURE SCRIPT "rqmblue2.sql" TARGET SCRIPT "rqmred2.sql";
SET LOG "bidir4.err";
SET RUN SCRIPT LATER;

# 2 Creating a replication queue map
The CREATE REPLQMAP command specifies an administration queue and receive queue within the queue manager QM1 that is used for SAMPLE, and a send queue within the queue manager QM2 that is used for SAMPLE2.

CREATE REPLQMAP SAMPLE2.BLUE_TO_SAMPLE.RED USING
ADMINQ "RED.QM2.ADMINQ" RECVQ "RED.QM2_TO_QM1.DATAQ"
SENDQ "BLUE.QM1_TO_QM2.DATAQ";

# 3 Ending the ASNCLP session
QUIT;

### ASNCLP script 5

This ASNCLP script contains commands for the Q subscriptions between the SAMPLE database and the SAMPLE2 database. You invoke this script by using the LOAD MULTIDIR REPL SCRIPT command in ASNCLP script 6. The script includes commands for the following tasks:

1. Setting the subgroup
2. Setting servers for the subgroup
3. Identifying the matching Q Capture and Q Apply schema at each server
Specifying replication queue maps to connect the servers in both directions
Specifying the table to be replicated (one copy at each server)
Creating the Q subscriptions

# No environment commands are required for this script. These commands are included
# in ASNCLP script 6, which invokes this script.

# 1 Setting the subgroup
SET SUBGROUP "bidirgroup";

# 2 Setting servers for the subgroup
SET SERVER MULTIDIR TO DBALIAS "SAMPLE";
SET SERVER MULTIDIR TO DBALIAS "SAMPLE2";

# 3 Identifying the matching schema of the Q Capture and Q Apply
# control tables at each server
SET MULTIDIR SCHEMA "SAMPLE",RED;
SET MULTIDIR SCHEMA "SAMPLE2",BLUE;

# 4 Specifying the replication queue maps that connect the two servers
# in both directions
SET CONNECTION SOURCE "SAMPLE",RED TARGET "SAMPLE2",BLUE REPLQMAP "SAMPLE_RED_TO_SAMPLE2_BLUE";
SET CONNECTION SOURCE "SAMPLE2",BLUE TARGET "SAMPLE",RED REPLQMAP "SAMPLE2_BLUE_TO_SAMPLE_RED";

# 5 Specifying the table to be replicated (one copy at each server)
# The SET TABLES command specifies only one table, RED.EMPLOYEE at the SAMPLE
# database. This prompts the command to generate SQL statements to create a matching
# table at the SAMPLE2 database, BLUE.TGTEMPLOYEE.
SET TABLES (SAMPLE,RED,RED.EMPLOYEE);

# 6 Creating the Q subscriptions
# The command uses two FROM NODE clauses to specify a conflict rule of C (check
# changed columns) and a conflict action of F (force changes into the target) for
# the SAMPLE database. For SAMPLE2 (the standby server), the conflict rule is
# A (check all columns) and the conflict action is I (ignore conflicts).
CREATE QSUB SUBTYPE B
FROM NODE SAMPLE,RED SOURCE ALL CHANGED ROWS Y HAS LOAD PHASE I
TARGET CONFLICT RULE C CONFLICT ACTION F
FROM NODE SAMPLE2,BLUE SOURCE ALL CHANGED ROWS N HAS LOAD PHASE E
TARGET CONFLICT RULE A CONFLICT ACTION I;

# No QUIT statement is required. The ASNCLP program reads this statement in
# script 6.

**ASNCLP script 6**

The final script uses the LOAD MULTIDIR REPL SCRIPT command to invoke
ASNCLP script 5 for creating the Q subscriptions. It includes commands for the
following tasks:

1. Setting the environment
2. Invoking the script that creates Q subscriptions
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates two SQL scripts that are automatically
# named after the databases, SAMPLE.sql and SAMPLE2.sql. Run each SQL script at the
# database for which it is named.
# IMPORTANT: Move or rename the existing SAMPLE.sql and SAMPLE2.sql scripts that were generated for creating control tables, or the statements for creating Q subscriptions will be appended to the end of the files.

ASNCNLPSSESSION SET TO Q REPLICAATION;
SET OUTPUT MULTIDIR;
SET LOG "bidir5.err";
SET RUN SCRIPT LATER;

# 2 Invoking the script that creates Q subscriptions
# Before you run this script, save ASNCLP script 5 in a file, bidirqsubs.in.
LOAD MULTIDIR REPL SCRIPT "/home/files/asnclp/bidirqsubs.in";

# 3 Ending the ASNCLP session
QUIT;

Output of the scripts

In addition to the log files, this example produces eight SQL script files in the same directory where you run the ASNCLP program. Table 5 describes the files.

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE database.</td>
</tr>
<tr>
<td>SAMPLE2.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE2 database.</td>
</tr>
<tr>
<td>rqmred1.sql</td>
<td>Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE_RED_TO_SAMPLE2_BLUE.</td>
</tr>
<tr>
<td>rqmblue1.sql</td>
<td>Add definitions to the Q Apply control tables at SAMPLE2 for the replication queue map SAMPLE_RED_TO_SAMPLE2_BLUE.</td>
</tr>
<tr>
<td>rqmred2.sql</td>
<td>Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE2_BLUE_TO_SAMPLE_RED.</td>
</tr>
<tr>
<td>rqmblue2.sql</td>
<td>Add definitions to the Q Apply control tables at SAMPLE2 for the replication queue map SAMPLE2_BLUE_TO_SAMPLE_RED.</td>
</tr>
<tr>
<td>SAMPLE.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.</td>
</tr>
<tr>
<td>SAMPLE2.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE2.</td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers)

This sample contains six ASNCLP scripts for setting up a peer-to-peer Q replication environment with two servers. It includes Q Capture and Q Apply control tables at both servers, replication queue maps in both directions, and two peer-to-peer Q subscriptions.
The scenario for these samples involves two databases, SAMPLE and SAMPLPEER. One table, DEPARTMENT, will be replicated in both directions between the two databases. The Q Capture and Q Apply programs at the SAMPLE database have the schema GREEN. The two corresponding programs at the SAMPLPEER database have the schema MAGENTA.

The ASNCLP scripts create eight SQL scripts. Table 6 on page 73 below the sample describes each SQL script.

Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Q Capture and Q Apply control tables at SAMPLE
2. Q Capture and Q Apply control tables at SAMPLPEER
3. Replication queue map from SAMPLE to SAMPLPEER
4. Replication queue map from SAMPLPEER to SAMPLE
5. Q subscriptions

You can copy the commands for each ASNCLP script to a text file, modify the values, and run the script by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

For help creating the WebSphere MQ objects that are used in these scripts, see
Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing and WebSphere MQ setup scripts for Q replication

ASNCLP script 1

This script creates control tables at the SAMPLE database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at SAMPLE
3. Ending the ASNCLP session
   # 1 Setting the environment
   # The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLE.sql, that is
   # automatically named after the database. The script contains SQL statements to
   # create both Q Capture and Q Apply control tables.
   # The SET LOG command directs ASNCLP messages to one log file, p2p2-log1.err.
   # The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
   # they are run.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p2-log1.err";
SET RUN SCRIPT LATER;

# 2 Creating Q Capture and Q Apply control tables at SAMPLE
# To use the script, change the ID and PASSWORD values.
# Both the Q Capture and Q Apply control tables will have the schema GREEN.

SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "password";
SET CAPTURE SCHEMA SOURCE GREEN;
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "GREEN.QM1.RESTARTQ" ADMINQ "GREEN.QM1.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "password";
SET APPLY SCHEMA GREEN;
SET QMANAGER "QM1" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# 3 Ending the ASNCLP session
QUIT;

**ASNCLP script 2**

This script creates control tables at the SAMPLPEER database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at SAMPLPEER
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLPEER.sql. The script contains SQL statements to create both Q Capture and Q Apply control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p2-log2.err";
SET RUN SCRIPT LATER;

# 2 Creating Q Capture and Q Apply control tables at SAMPLPEER
# Both the Q Capture and Q Apply control tables will have the schema MAGENTA.

SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE GREEN;
SET QMANAGER "QM2" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "MAGENTA.QM2.RESTARTQ" ADMIXQ "MAGENTA.QM2.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLPEER ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA MAGENTA;
SET QMANAGER "QM2" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# 3 End the ASNCLP session.
QUIT;

**ASNCLP script 3**

This script creates a replication queue map from SAMPLE to SAMPLPEER. It includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map from SAMPLE to SAMPLPEER
3. Ending the ASNCLP session

# 1 Setting the environment
# Two SET SERVER commands are required because the replication queue map from SAMPLE to SAMPLPEER is defined in the Q Capture control tables at SAMPLE and the Q Apply control tables at SAMPLPEER.
# The SET OUTPUT command specifies two SQL scripts: rqmgreen1.sql, which adds definitions to SAMPLE, and rqmmagenta1.sql, which adds definitions to SAMPLPEER.

ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE GREEN;
SET SERVER TARGET TO DBALIAS SAMPLPEER ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA MAGENTA;
SET OUTPUT CAPTURE SCRIPT "rqmgreen1.sql" TARGET SCRIPT "rqmmagenta1.sql";
SET LOG "p2p2-log3.err";
SET RUN SCRIPT LATER;

# 2 Creating a replication queue map
# The CREATE REPLQMAP command specifies an administration queue and receive queue
# within the queue manager QM2 that is used for SAMPLPEER, and a send queue within
# the queue manager QM1 that is used for SAMPLE.
CREATE REPLQMAP SAMPLE_Green_TO_SAMPLPEER_MAGENTA USING
ADMINQ "MAGENTA.QM1.ADMINQ" RECVQ "MAGENTA.QM1_TO_QM2.DATAQ"
SENDQ "GREEN.QM1_TO_QM2.DATAQ";

# 3 Ending the ASNCLP session
QUIT;

ASNCLP script 4

This script creates a replication queue map from SAMPLPEER to SAMPLE. It includes commands for the following tasks:

1. Setting the environment
2. Creating a replication queue map from SAMPLPEER to SAMPLE
3. Ending the ASNCLP session

# 1 Setting the environment
# Two SET SERVER commands are required because the replication queue map from
# SAMPLPEER to SAMPLE is defined in the Q Capture control tables at SAMPLPEER
# and the Q Apply control tables at SAMPLE.
# The SET OUTPUT command specifies two SQL scripts, rqmmagenta2.sql, which adds
# definitions to SAMPLPEER, and rqmgreen1.sql, which adds definitions to SAMPLE.
ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO DBALIAS SAMPLPEER ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE MAGENTA;
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA GREEN;
SET OUTPUT CAPTURE SCRIPT "rqmmagenta2.sql" TARGET SCRIPT "rqmgreen2.sql";
SET LOG "p2p2-log4.err";
SET RUN SCRIPT LATER;

# 2 Creating a replication queue map
# The CREATE REPLQMAP command specifies an administration queue and receive
# queue within the queue manager QM1 that is used for SAMPLE, and a send queue
# within the queue manager QM2 that is used for SAMPLPEER.
CREATE REPLQMAP SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN USING
ADMINQ "GREEN.QM2.ADMINQ" RECVQ "GREEN.QM2_TO_QM1.DATAQ"
SENDQ "MAGENTA.QM1_TO_QM2.DATAQ";

# 3 Ending the ASNCLP session
QUIT;

ASNCLP script 5

This ASNCLP script contains commands for the Q subscriptions between the
SAMPLE database and the SAMPLPEER database. You invoke this script by using
the LOAD MULTIDIR REPL SCRIPT command in ASNCLP script 6. The script
includes commands for the following tasks:

1. Setting the subgroup
2. Setting servers for the subgroup
3. Identifying the matching Q Capture and Q Apply schema at each server
Specifying queue maps that connect the servers in both directions
Specifying the table to be replicated (one copy at each server)
Creating the Q subscriptions

Specifying the table to be replicated (one copy at each server)

Creating the Q subscriptions

# No environment commands are required for this script. These commands are
# included in ASNCLP script 6, which invokes this script.

# 2 Setting the subgroup

SET SUBGROUP "p2p2group";

# 3 Setting servers for the subgroup

SET SERVER MULTIDIR TO DBALIAS "SAMPLE";
SET SERVER MULTIDIR TO DBALIAS "SAMPLPEER";

# 4 Identifying the matching schema of the Q Capture and Q Apply
# control tables at each server

SET MULTIDIR SCHEMA "SAMPLE".GREEN;
SET MULTIDIR SCHEMA "SAMPLPEER".MAGENTA;

# 5 Specifying the replication queue maps that connect the two servers
# in both directions

SET CONNECTION SOURCE "SAMPLE".GREEN TARGET "SAMPLPEER".MAGENTA REPLQMAP
"SAMPLE_GREEN_TO_SAMPLPEER_MAGENTA";
SET CONNECTION SOURCE "SAMPLPEER".MAGENTA TARGET "SAMPLE".GREEN REPLQMAP
"SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN";

# 6 Specifying the table to be replicated (one copy at each server)
# The SET TABLES command specifies only one table, GREEN.DEPARTMENT at the
# SAMPLE database. This prompts the command to generate SQL statements to create
# a matching table at the SAMPLPEER database, MAGENTA.TGTEMPLOYEE.

SET TABLES (SAMPLE.GREEN.DEPARTMENT);

# 7 Creating the Q subscriptions
# A single CREATE QSUB command generates commands to create two peer-to-peer
# Q subscriptions between SAMPLE and SAMPLPEER.

CREATE QSUB SUBTYPE P;

# No QUIT statement is required. The ASNCLP program reads this statement in
# script 6.

**ASNCLP script 6**

The final script uses the LOAD MULTIDIR REPL SCRIPT command to invoke
ASNCLP script 5 for creating the Q subscriptions. It includes commands for the
following tasks:

1 Setting the environment
2 Invoking the script that creates Q subscriptions
3 Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates two SQL scripts that are automatically
# named after the databases, SAMPLE.sql and SAMPLPEER.sql. Run each SQL script
# at the database for which it is named.
# IMPORTANT: Move or rename the existing SAMPLE.sql and SAMPLPEER.sql scripts
# that were generated for creating control tables, or the statements for creating
# Q subscriptions will be appended to the end of the files.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p2-log5.err";
SET RUN SCRIPT LATER;

# 2 Invoking the script that creates Q subscriptions
# Before you run this script, save ASNCLP script 5 in a file, p2p2qsubs.in.

LOAD MULTIDIR REPL SCRIPT "/home/files/asnclp/p2p2qsubs.in";

# 3 Ending the ASNCLP session
QUIT;

Output of the scripts

In addition to the log files, this example produces eight SQL script files in the
same directory where you run the ASNCLP program. Table 6 describes the files.

Table 6. SQL script files that are created by the sample ASNCLP scripts

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE database.</td>
</tr>
<tr>
<td>SAMPLPEER.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLPEER database.</td>
</tr>
<tr>
<td>rqmgreen1.sql</td>
<td>Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE_GREEN_TO_SAMPLPEER_MAGENTA.</td>
</tr>
<tr>
<td>rqmmagenta1.sql</td>
<td>Add definitions to the Q Apply control tables at SAMPLPEER for the replication queue map SAMPLE_GREEN_TO_SAMPLPEER_MAGENTA.</td>
</tr>
<tr>
<td>rqmgreen2.sql</td>
<td>Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN.</td>
</tr>
<tr>
<td>rqmmagenta2.sql</td>
<td>Add definitions to the Q Apply control tables at SAMPLPEER for the replication queue map SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN.</td>
</tr>
<tr>
<td>SAMPLE.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.</td>
</tr>
<tr>
<td>SAMPLPEER.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLPEER.</td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up peer-to-peer Q replication (three servers)

This sample contains six ASNCLP scripts for setting up peer-to-peer Q replication with three servers. It includes Q Capture and Q Apply control tables at each of the three servers, replication queue maps in both directions between each server, and six Q subscriptions between the servers.

The scenario involves three databases, SAMPLE, SAMPLE2, and SAMPLE3. One table, STAFF, will be replicated between the three databases. The Q Capture and Q
Apply programs at the SAMPLE database have the schema GRAY. At SAMPLE2 the programs have the schema BROWN, and at SAMPLE3 the schema is YELLOW.

The ASNCLP scripts create 12 SQL scripts. Table 7 on page 79 below the sample describes each SQL script.

Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Q Capture and Q Apply control tables at SAMPLE
2. Q Capture and Q Apply control tables at SAMPLE2
3. Q Capture and Q Apply control tables at SAMPLE3
4. Replication queue maps
5. Q subscriptions

You can copy the commands for each ASNCLP script to a text file, modify the values, and run the script by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

For help creating the WebSphere MQ objects that are used in these scripts, see Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing and WebSphere MQ setup scripts for Q replication.

**ASNCLP script 1**

This script creates control tables at the SAMPLE database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at the SAMPLE database
3. Ending the ASNCLP session

Setting the environment

The SET OUTPUT MULTIDIR command creates one SQL script that is automatically named after the database, SAMPLE.sql. The script contains SQL statements to create both Q Capture and Q Apply control tables.

The SET LOG command directs ASNCLP messages to one log file, p2p3-log1.err. The SET RUN SCRIPT LATER option allows you to review the SQL scripts before they are run.

ASNCLP SESSION SET TO Q REPLICACTION;
SET OUTPUT MULTIDIR;
SET LOG "p2p3-log1.err";
SET RUN SCRIPT LATER;

Creating Q Capture and Q Apply control tables at SAMPLE

To use the script, change the ID and PASSWORD values. Both the Q Capture and Q Apply control tables will have the schema GRAY.

SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE GRAY;
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "GRAY.QM1.RESTARTQ" ADMINQ "GRAY.QM1.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA GRAY;
SET QMANAGER "QM1" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";
# 3 Ending the ASNCLP session
QUIT;

**ASNCLP script 2**

This script creates control tables at the SAMPLE2 database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at the SAMPLE2 database
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLE2.sql.
# The script contains SQL statements to create both Q Capture and Q Apply control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p3-log2.err";
SET RUN SCRIPT LATER;

# 2 Creating Q Capture and Q Apply control tables at SAMPLE2
# Both the Q Capture and Q Apply control tables will have the schema BROWN.

SET SERVER CAPTURE TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE BROWN;
SET QMANAGER "QM2" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "BROWN.QM2.RESTARTQ" ADMINQ "BROWN.QM2.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA BROWN;
SET QMANAGER "QM2" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# 3 End the ASNCLP session.
QUIT;

**ASNCLP script 3**

This script creates control tables at the SAMPLE3 database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture and Q Apply control tables at the SAMPLE3 database
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates one SQL script, SAMPLE3.sql.
# The script contains SQL statements to create both Q Capture and Q Apply control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p3-log3.err";
SET RUN SCRIPT LATER;

# 2 Creating Q Capture and Q Apply control tables at SAMPLE3
# Both the Q Capture and Q Apply control tables will have the schema YELLOW.

SET SERVER CAPTURE TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE YELLOW;
SET QMANAGER "QM3" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "YELLOW.QM3.RESTARTQ" ADMINQ "YELLOW.QM3.ADMINQ";
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA YELLOW;
SET QMANAGER "QM3" FOR APPLY SCHEMA;
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut";

# End the ASNCLP session.
QUIT;

**ASNCLP script 4**

This script creates six replication queue maps, from SAMPLE to SAMPLE2 and SAMPLE3, from SAMPLE2 to SAMPLE and SAMPLE3, and from SAMPLE3 to SAMPLE and SAMPLE2:

1. Creating the replication queue maps
2. Ending the ASNCLP session

# Creating the replication queue maps
# Two servers and two schemas need to be set for each CREATE REPLQMAP command.
# The SET OUTPUT commands create a separate SQL script for each queue map.

ASNCLP SESSION SET TO Q REPLICATON;
SET LOG "p2p3-log4.err";
SET RUN SCRIPT LATER;

# First queue map
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE GRAY;
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA BROWN;
SET OUTPUT CAPTURE SCRIPT "rqm1.sql" TARGET SCRIPT "rqm1.sql";
CREATE REPLQMAP SAMPLE2_GRAY_TO_SAMPLE2_BROWN USING
ADMINQ "BROWN.QM2.ADMINQ" RECVQ "GRAY.QM1_TO_QM2.DATAQ"
SENDQ "GRAY.QM1_TO_QM2.DATAQ";

# Second queue map
SET SERVER CAPTURE TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE BROWN;
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA GRAY;
SET OUTPUT CAPTURE SCRIPT "rqm2.sql" TARGET SCRIPT "rqm2.sql";
CREATE REPLQMAP SAMPLE2_BROWN_TO_SAMPLE_GRAY USING
ADMINQ "GRAY.QM2.ADMINQ" RECVQ "GRAY.QM2_TO_QM1.DATAQ"
SENDQ "BROWN.QM2_TO_QM1.DATAQ";

# Third queue map
# The SET SERVER CAPTURE and SET CAPTURE SCHEMA commands for SAMPLE2.BROWN
# are still in effect.
SET SERVER TARGET TO DBALIAS SAMPLE3 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA YELLOW;
SET OUTPUT CAPTURE SCRIPT "rqm3.sql" TARGET SCRIPT "rqm3.sql";
CREATE REPLQMAP SAMPLE2_BROWN_TO_SAMPLE3_YELLOW USING
ADMINQ "YELLOW.QM3.ADMINQ" RECVQ "YELLOW.QM2_TO_QM3.DATAQ"
SENDQ "BROWN.QM2_TO_QM3.DATAQ";

# Fourth queue map
SET SERVER CAPTURE TO DBALIAS SAMPLE3 ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE YELLOW;
SET SERVER TARGET TO DBALIAS SAMPLE2 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA BROWN;
SET OUTPUT CAPTURE SCRIPT "rqm4.sql" TARGET SCRIPT "rqm4.sql";
CREATE REPLQMAP SAMPLE3_YELLOW_TO_SAMPLE2_BROWN USING
ADMINQ "BROWN.QM2.ADMINQ" RECVQ "BROWN.QM3_TO_QM2.DATAQ"
SENDQ "YELLOW.QM3_TO_QM2.DATAQ";

# Fifth queue map
# The SET SERVER CAPTURE and SET CAPTURE SCHEMA commands for SAMPLE3.YELLOW
# are still in effect.
SET SERVER TARGET TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA GRAY;
SET OUTPUT CAPTURE SCRIPT "rqm5.sql" TARGET SCRIPT "rqm5.sql";
CREATE REPLQMAP SAMPLE3.YELLOW TO SAMPLEGRAY USING
ADMINQ "GRAY.QM1.ADMINQ" RECVQ "GRAY.QM3_TO_QM1.DATAQ"
SENDQ "YELLOW.QM3_TO_QM1.DATAQ";

# Sixth queue map
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE GRAY;
SET SERVER TARGET TO DBALIAS SAMPLE3 ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA YELLOW;
SET OUTPUT CAPTURE SCRIPT "rqm6.sql" TARGET SCRIPT "rqm6.sql";
CREATE REPLQMAP SAMPLEGRAY TO SAMPLE3.YELLOW USING
ADMINQ "YELLOW.QM3.ADMINQ" RECVQ "YELLOW.QM1_TO_QM3.DATAQ"
SENDQ "GRAY.QM3_TO_QM1.DATAQ";

# Ending the ASNCLP session
QUIT;

**ASNCLP script 5**

This ASNCLP script contains commands for the Q subscriptions between the three servers. You invoke this script by using the LOAD MULTIDIR REPL SCRIPT command in ASNCLP script 6. The script includes commands for the following tasks:

1. Setting the subgroup
2. Setting servers for the subgroup
3. Identifying the matching Q Capture and Q Apply schema at each server
4. Specifying the replication queue maps that connect the three servers
5. Specifying the table to be replicated (one copy at each server)
6. Creating the Q subscriptions

# No environment commands are required for this script. These commands are included
# in ASNCLP script 6, which invokes this script.

# Setting the subgroup
SET SUBGROUP "p2p3group";

# Setting servers for the subgroup
SET SERVER MULTIDIR TO DBALIAS "SAMPLE";
SET SERVER MULTIDIR TO DBALIAS "SAMPLE2";
SET SERVER MULTIDIR TO DBALIAS "SAMPLE3";

# Identifying the matching schema of the Q Capture and Q Apply
# control tables at each server
SET MULTIDIR SCHEMA "SAMPLE".GRAY;
SET MULTIDIR SCHEMA "SAMPLE2".BROWN;
SET MULTIDIR SCHEMA "SAMPLE3".YELLOW;

# Specifying the replication queue maps that connect the two servers
# in both directions
SET CONNECTION SOURCE "SAMPLE".GRAY TARGET "SAMPLE2".BROWN REPLQMAP
"SAMPLE_GRAY_TO_SAMPLE2_BROWN";
SET CONNECTION SOURCE "SAMPLE".GRAY TARGET "SAMPLE3".YELLOW REPLQMAP
"SAMPLE_GRAY_TO_SAMPLE3_YELLOW";
SET CONNECTION SOURCE "SAMPLE2".BROWN TARGET "SAMPLE".GRAY REPLQMAP
"SAMPLE2_BROWN_TO_SAMPLE_GRAY";
SET CONNECTION SOURCE "SAMPLE2".BROWN TARGET "SAMPLE3".YELLOW REPLQMAP
"SAMPLE2_BROWN_TO_SAMPLE3_YELLOW";
SET CONNECTION SOURCE "SAMPLE3".YELLOW TARGET "SAMPLE".GRAY REPLQMAP
"SAMPLE3_YELLOW_TO_SAMPLE_GRAY";
SET CONNECTION SOURCE "SAMPLE3".YELLOW TARGET "SAMPLE2".BROWN REPLQMAP
"SAMPLE3_YELLOW_TO_SAMPLE2_BROWN";

# 6 Specifying the table to be replicated (one copy at each server)
# The SET TABLES command specifies only one table, GRAY.STAFF at the SAMPLE
# database. This prompts the command to generate SQL statements to create a
# matching table at the SAMPLE2 database, BROWN.TGTSTAFF, and at the SAMPLE3
# database, YELLOW.TGTSTAFF.

SET TABLES (SAMPLE.GRAY.GRAY.STAFF);

# 7 Creating the Q subscriptions
# A single CREATE QSUB command generates commands to create six peer-to-peer
# Q subscriptions between SAMPLE, SAMPLE2, and SAMPLE3.

CREATE QSUB SUBTYPE P;

# No QUIT statement is required. The ASNCLP program reads this statement in
# script 5.

**ASNCLP script 6**

The final script uses the LOAD MULTIDIR REPL SCRIPT command to invoke
ASNCLP script 5 for creating the Q subscriptions. It includes commands for the
following tasks:

1. Setting the environment
2. Invoking the script that creates Q subscriptions
3. Ending the ASNCLP session

# 1 Setting the environment
# The SET OUTPUT MULTIDIR command creates three SQL scripts that are
# automatically named after the databases, SAMPLE.sql, SAMPLE2.sql, and
# SAMPLE3.sql. Run each SQL script at the database for which it is named.
# IMPORTANT: Move or rename the existing SAMPLE.sql, SAMPLE2.sql, and
# SAMPLE3.sql scripts that were generated for creating control tables,
# or the statements for creating Q subscriptions will be appended
# to the end of the files.

ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p3-log5.err";
SET RUN SCRIPT LATER;

# 2 Invoking the script that creates Q subscriptions
# Before you run this script, save ASNCLP script 5 in a file, p2p3qsubs.in.

LOAD MULTIDIR REPL SCRIPT "/home/files/asnclp/p2p3qsubs.in";

# 3 Ending the ASNCLP session

QUIT;
Output of the scripts

In addition to the log files, this example produces 12 SQL script files in the same directory where you run the ASNCLP program. Table 7 describes the files.

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE database.</td>
</tr>
<tr>
<td>SAMPLE2.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE2 database.</td>
</tr>
<tr>
<td>SAMPLE3.sql</td>
<td>Create Q Capture and Q Apply control tables at the SAMPLE2 database.</td>
</tr>
<tr>
<td>rqm1.sql</td>
<td>Define the queue map SAMPLE_GRAY_TO_SAMPLE2_BROWN</td>
</tr>
<tr>
<td>rqm2.sql</td>
<td>Define the queue map SAMPLE2_BROWN_TO_SAMPLE_GRAY</td>
</tr>
<tr>
<td>rqm3.sql</td>
<td>Define the queue map SAMPLE2_BROWN_TO_SAMPLE3_YELLOW</td>
</tr>
<tr>
<td>rqm4.sql</td>
<td>Define the queue map SAMPLE3_YELLOW TO SAMPLE2_BROWN</td>
</tr>
<tr>
<td>rqm5.sql</td>
<td>Define the queue map SAMPLE3_YELOW TO SAMPLE_GRAY</td>
</tr>
<tr>
<td>rqm6.sql</td>
<td>Define the queue map SAMPLE_GRAY TO SAMPLE3_YELOW</td>
</tr>
<tr>
<td>SAMPLE.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.</td>
</tr>
<tr>
<td>SAMPLE2.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE2.</td>
</tr>
<tr>
<td>SAMPLE3.sql</td>
<td>Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE3.</td>
</tr>
</tbody>
</table>

Sample ASNCLP script for promoting unidirectional configurations

This sample contains an ASNCLP script for promoting a unidirectional Q replication configuration. You can copy an existing Q replication or event publishing configuration to another system by promoting that configuration by using a set of ASNCLP scripts. These commands scan and discover the Q replication control table and DB2 catalog table on specified source servers, and then create replication definitions. You can execute scripts containing these definitions on any destination server to recreate the replication environment there.

You can customize some of the properties of the destination environment.
Suppose you want to promote a replication environment you have created on a test server configuration to your production server configuration. The test configuration consists of Q Capture server TESTCAP and Q Apply server TESTAPP, with the following details:

- Q Capture control tables on server TESTCAP exist under schema ASN
- Q Apply control tables on server TESTAPP exist under schema ASN
- 10 replication queue maps exist between the servers, named qmap1 to qmap10
- 30 Q subscriptions exist on each queue map
- A total of 300 unidirectional Q subscriptions exist between these servers

To promote all replication queue maps and all Q subscriptions that use them from the test environment to the production one, create the following ASNCLP input script:

```
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG promote-repqmap-qsub.log;

SET SERVER CAPTURE TO DBALIAS TESTCAP ID id1 PASSWORD "p1"
    PROMOTE TO DBALIAS PRODCAP ID id1 PASSWORD "plwd" SCHEMA ASN;

SET SERVER TARGET TO DBALIAS TESTAPP ID id1 PASSWORD "plwd"
    PROMOTE TO DBALIAS PRODCAP ID id1 PASSWORD "plwd" SCHEMA ASN;

#This is the output script that will be generated by these commands
SET OUTPUT PROMOTE SCRIPT "replqmap_qsub.in";

#These two statements will be put in the generated script
SET OUTPUT CAPTURE SCRIPT "promote_capture_repqmap.sql";
SET OUTPUT TARGET SCRIPT "promote_target_repqmap.sql";

#Generate ASNCLP commands for promoting all queue maps that match this predicate
    PROMOTE REPLQMAP LIKE "qmap%";

#Generate ASNCLP commands for promoting all Q subscriptions that use these queue maps
    PROMOTE QSUB FOR REPLQMAP LIKE "qmap%";
```

The output of this script is another ASNCLP script that is named replqmap_qsub.in which includes the command SET RUN SCRIPT NOW.

Running this script by using asnclp -f "asnclp_replqmap.in" executes the SQL definitions and persists the information in the control tables, promoting the specified environment.

**Sample ASNCLP scripts for promoting peer-to-peer configurations**

This sample contains three ASNCLP scripts for promoting a peer-to-peer Q replication configuration. You can copy an existing Q replication or event publishing configuration to another system by promoting that configuration by using a set of ASNCLP scripts. These commands scan and discover the Q replication control table and DB2 catalog table on the specified source servers, and then create replication definitions. You can execute scripts containing these definitions on any destination server to recreate the replication environment there.

The scenario for these samples involves an existing configuration with peer-to-peer Q subscriptions between server SAMPLE, schema ASN and server TESTDB, schema BSN.
• Q Capture and Q Apply control tables exist on server SAMPLE under schema ASN, and on server TESTDB under schema BSN
• Two replication queue maps exist between SAMPLE.ASN and TESTDB.BSN:
  – RQMap1 includes send queue SQ1, receive queue RQ1, and administration queue AQ1.
  – RQMap2 includes send queue SQ2, receive queue RQ2, and administration queue AQ2.

The sample scripts promote existing objects for the peer-to-peer configuration to server SAMPLE.ASN1 and server TESTDB1.BSN1. The scripts promote both replication queue maps and all Q subscriptions that use these queue maps.

Create the control tables on destination servers

These scripts assume that you first created Q Capture and Q Apply control tables on the promote destination: on server SAMPLE under schema ASN1 and on server TESTDB1 under schema BSN1. Create the control tables using the CREATE CONTROL TABLES FOR command or the Replication Center.

Promote the first replication queue map

This script promotes the replication queue map that moves data from the first peer to the second peer.
```plaintext
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG promote_repqmaplog;

#Identify the first peer's Q Capture and the second peer's Q Apply. The promote-to passwords are added to the generated script, but no connect is issued to the promote-to servers until the generated script is run.

#Identify the Q Capture server for the first peer and the Q Apply server for the second peer. The specified promote-to passwords are added to the generated script so that it can successfully execute. This connection information is only used when the generated script in the output script file is run.

SET SERVER CAPTURE TO DBALIAS SAMPLE ID id1 PASSWORD "p1"
PROMOTE TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd" SCHEMA ASN1;

SET SERVER TARGET TO DBALIAS TESTDB ID id1 PASSWORD "p1wd"
PROMOTE TO DBALIAS TESTDB1 ID id1 PASSWORD "p1wd" SCHEMA BSN1;

#This command defines the file that contains the output script that is generated.
SET OUTPUT PROMOTE SCRIPT "repqmap.in";

#These two SET OUTPUT statements are put in the generated script.
SET OUTPUT CAPTURE SCRIPT "promote_capture_repqmap.sql"
SET OUTPUT TARGET SCRIPT "promote_target_repqmap.sql"

#Generate the ASNCLP commands for promoting the replication queue map that is named RQMap1.
PROMOTE REPLQMAP NAME RQMap1;
```

Promote the second replication queue map

This script promotes the replication queue map that moves data from the second peer to the first peer.
```plaintext
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG promote_repqmaplog;
```
#Identify the Q Capture server for the second peer and the Q Apply server for the
#first peer. The specified promote-to passwords are added to the generated script
#so that it can successfully execute. This connection information is only used
#when the generated script in the output script file is run.

SET SERVER CAPTURE TO DBALIAS TESTDB ID id1 PASSWORD "p1"
PROMOTE TO DBALIAS TESTDB1 ID id1 PASSWORD "p1wd" SCHEMA BSN1;

SET SERVER TARGET TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd"
PROMOTE TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd" SCHEMA ASN1;

#This command defines the file that contains the output script that is generated.
SET OUTPUT PROMOTE SCRIPT "repopmap.in";

#These two SET OUTPUT statements are put in the generated script
SET OUTPUT CAPTURE SCRIPT "promote_capture_repopmap.sql";
SET OUTPUT TARGET SCRIPT "promote_target_repopmap.sql";

#Generate the ASNCLP commands for promoting the replication queue
#map that is named RQMap2
PROMOTE REPLQMAP NAME RQMap2;

Run the generated scripts

Run the generated ASNCLP scripts by using the asnclp -f repqmap.in command
from a system command prompt. Run the SQL output that is generated by these
scripts.

Promote the Q subscriptions

Promote the peer-to-peer Q subscriptions for the replication queue maps:
SET PEER NODE 1 SERVER DBALIAS SAMPLE ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd" SCHEMA ASN1;

SET PEER NODE 2 SERVER DBALIAS TESTDB ID id1 PASSWORD "p1wd" SCHEMA BSN
PROMOTE TO DBALIAS TESTDB1 ID id1 PASSWORD "p1wd" SCHEMA BSN1;

#Generate the ASNCLP scripts to promote all Q subscriptions that use
#replication queue maps with names that begin with the predicate RQMAP:
PROMOTE QSUB REPLQMAP LIKE "RQMAP*";

 ASNCLP commands for unidirectional Q replication

The ASNCLP commands for unidirectional Q replication set the environment,
define, change, and delete Q subscriptions, and specify output files. Some of the
ASNCLP commands for unidirectional replication also apply to Classic replication.

```
"Sample ASNCLP scripts for setting up unidirectional Q replication" on page 57
and "Sample ASNCLP scripts for setting up unidirectional Q replication from a
Classic data source" on page 61 demonstrate how you can combine ASNCLP
commands to create an ASNCLP setup script.

Table 8 lists the ASNCLP commands for unidirectional Q replication and links to
topics that describe each command.
```

Table 8. ASNCLP commands for unidirectional Q replication

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a column to a Q subscription</td>
<td>ALTER ADD COLUMN command</td>
</tr>
<tr>
<td>If you want to ...</td>
<td>Use this command</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Update the IBMQREP_CAPPARMS table when you replicate from a Classic source</td>
<td>“ALTER CAPTURE PARAMETERS command (Classic replication)” on page 84</td>
</tr>
<tr>
<td>Change a Q subscription</td>
<td>“ALTER QSUB command (unidirectional replication)” on page 85</td>
</tr>
<tr>
<td>Change a replication queue map</td>
<td>ALTERN REPLQMAP command</td>
</tr>
<tr>
<td>Establish a session for Q replication</td>
<td>ASNCLP SESSION SET TO command</td>
</tr>
<tr>
<td>Create a Q subscription</td>
<td>“CREATE QSUB command (unidirectional replication)” on page 89</td>
</tr>
<tr>
<td>Create the control tables for the Q Capture and Q Apply programs</td>
<td>CREATE CONTROL TABLES FOR command</td>
</tr>
<tr>
<td>Create a replication queue map</td>
<td>CREATE REPLQMAP command</td>
</tr>
<tr>
<td>Delete a Q subscription</td>
<td>“DROP QSUB command (unidirectional Q replication)” on page 102</td>
</tr>
<tr>
<td>Drop the control tables for the Q Capture and Q Apply programs</td>
<td>DROP CONTROL TABLES ON command</td>
</tr>
<tr>
<td>Delete a replication queue map</td>
<td>DROP REPLQMAP command</td>
</tr>
<tr>
<td>List Q subscriptions</td>
<td>“LIST QSUB command (Q replication)” on page 103</td>
</tr>
<tr>
<td>List replication queue maps</td>
<td>“LIST REPLQMAP command (Q replication)” on page 105</td>
</tr>
<tr>
<td>List Q Apply schemas</td>
<td>LIST APPLY SCHEMA command</td>
</tr>
<tr>
<td>List Q Capture schemas</td>
<td>LIST CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Signal that a manual load of the target table is complete</td>
<td>LOAD DONE command</td>
</tr>
<tr>
<td>Promote a Q subscription</td>
<td>PROMOTE QSUB command</td>
</tr>
<tr>
<td>Promote a replication queue map</td>
<td>PROMOTE REPLQMAP command</td>
</tr>
<tr>
<td>• Specify whether to drop the target table when you delete a Q subscription</td>
<td>“SET DROP command (unidirectional replication)” on page 106</td>
</tr>
<tr>
<td>• Specify whether to drop the table space when you drop the target table or control tables</td>
<td></td>
</tr>
<tr>
<td>Set the Q Apply schema for all task commands</td>
<td>SET APPLY SCHEMA command</td>
</tr>
<tr>
<td>Set the Q Capture schema for all task commands</td>
<td>SET CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Define the log file for the ASNCLP program</td>
<td>SET LOG command</td>
</tr>
<tr>
<td>Define output files that contain SQL statements to set up unidirectional Q replication</td>
<td>SET OUTPUT command</td>
</tr>
<tr>
<td>Specify custom parameters for database objects to be created implicitly</td>
<td>SET PROFILE command</td>
</tr>
<tr>
<td>Set the WebSphere MQ queue manager</td>
<td>SET QMANAGER command</td>
</tr>
<tr>
<td>Specify whether to automatically run each task command from an input file before the ASNCLP program processes the next task command</td>
<td>SET RUN SCRIPT command</td>
</tr>
<tr>
<td>Specify the Q Capture server or Q Apply server to use in the ASNCLP session for unidirectional replication.</td>
<td>SET SERVER command</td>
</tr>
<tr>
<td>Enable and disable the trace for the ASNCLP commands</td>
<td>SET TRACE command</td>
</tr>
<tr>
<td>Display the environment set during the session</td>
<td>SHOW SET ENV command</td>
</tr>
<tr>
<td>If you want to ...</td>
<td>Use this command</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Start a Q subscription</td>
<td><strong>START QSUB command</strong></td>
</tr>
<tr>
<td>Stop a Q subscription</td>
<td><strong>STOP QSUB command</strong></td>
</tr>
<tr>
<td>Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions</td>
<td><strong>VALIDATE WSMQ ENVIRONMENT FOR command</strong></td>
</tr>
<tr>
<td>Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map</td>
<td><strong>VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command</strong></td>
</tr>
</tbody>
</table>

**ALTER CAPTURE PARAMETERS command (Classic replication)**

The capture operational parameters are stored in the table IBMQREP_CAPPARMS table. Use the ALTER CAPTURE PARAMETERS command to update the IBMQREP_CAPPARMS table when you replicate from a Classic source.

**Syntax**

```
ALTER CAPTURE PARAMETERS QMGR qmgr RESTARTQ restartq ADMINQ adminq
```

**Parameters**

QMGR `qmgr`

Specifies the queue manager name.

RESTARTQ `restartq`

Specifies the name of the restart queue that is used by the publication service.

ADMINQ `adminq`

Specifies the name of the administration queue that is used by the publication service.

**Usage notes**

- Issue this command before you define replication objects that interact with Classic data sources. Other commands that create and manipulate replication objects will not work properly if a row does not exist in the IBMQREP_CAPPARMS table.

**Example**

The following ALTER CAPTURE PARAMETERS command specifies the queue manager, restart queue, and administration queue for a Classic data source.

```
ASNCLP SESSION SET TO Q REPLICATION
SET SERVER CAPTURE CONFIG SERVER classic1
FILE asnservers.ini ID id1 PASSWORD passwd1;
ALTER CAPTURE PARAMETERS QMGR qmg1 RESTARTQ rq1 ADMINQ aq1;
```

**ALTER CONFIGURATION APPLY command**

The ALTER CONFIGURATION APPLY command allows you to change the configuration of the Q Apply program after you have specified a target server and Q Apply schema.
Syntax

```
ALTER CONFIGURATION APPLY SET CAPTURE SCHEMA
```

Parameters

```
set"name"
```

Specifies the new SQL Capture schema for the registrations of the CCD tables that Q apply maintains.

```
is null
```

Specifies that Q Apply does not maintain the registrations of its target CCD tables.

Usage notes

- Use this command in order to configure a Q Apply program to manage a SQL Capture schema.

Example

This example specifies that Q Apply uses the capture schema "ASN".

```
ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER TARGET TO QAPPDB;
SET APPLY SCHEMA QAPP1;
ALTER CONFIGURATION APPLY SQL CAPTURE SCHEMA SET "ASN";
```

**ALTER QSUB command (unidirectional replication)**

Use the ALTER QSUB command to change the properties of a Q subscription for unidirectional Q replication.

Syntax

```
ALTER QSUB—subname—REPLQMAP—mapname—USING REPLQMAP—mapname—DESC—description—
MANAGE TARGET CCD—action—USING OPTIONS—other-opt-clause—
```

action:

```
CREATE SQL REGISTRATION
DROP SQL REGISTRATION
ALTER SQL REGISTRATION FOR Q REPLICATION
```

other-opt-clause:

```
SEARCH CONDITION—"search_condition"—ALL CHANGED ROWS—N
```
Parameters

**QSUB subname**
- Specifies the name of the Q subscription.

**REPLQMAP mapname**
- Specifies the name of the replication queue map for the Q subscription.

**USING REPLQMAP mapname**
- Specify to alter the Q subscription and to use a different replication queue map.

**DESC description**
- Specifies a description of the Q subscription.

**action:**

**CREATE SQL REGISTRATION**
- Registers the target CCD table for the Q subscription as a source for SQL replication.

**DROP SQL REGISTRATION**
- Deletes an existing registration for SQL replication. When you issue the CREATE QSUB command with this parameter, the ASNCLP checks to make sure that all Q subscriptions that use this registration are inactive.

**ALTER SQL REGISTRATION FOR Q REPLICATION**
- Modifies an existing registration for SQL replication by updating the CD_OWNER field in the IBMSNAP_REGISTER table with the Q Apply schema and the CD_TABLE field with the name of the receive queue for the Q subscription. You can also use this action to change an existing SQL registration to a Q subscription that uses a different receive queue.

**other-opt-clause:**

**SEARCH CONDITION "search_condition"**
- Specifies a search condition for filtering changes to replicate. You cannot use...
this parameter with Classic sources. The change is not sent if the predicate is false. This is an annotated select WHERE clause, where there must be a colon before the column names of the table to be replicated. The following example shows a WHERE clause:

```
ALTER QSUB myqsub REPLQMAP replmap10 USING OPTIONS SEARCH CONDITION "WHERE :MYKEY > 1000"
```

**ALL CHANGED ROWS**

Specifies the data sending option.

- **N** Send a row only if a subscribed column in the source table changes.
- **Y** Send a row when any column in the source table changes.

**HAS LOAD PHASE**

Specifies whether the target table for the Q subscription will be loaded with data from the source.

- **N** No load phase at the target. This is the default.
- **I** Specifies an automatic load. The Q Apply program loads from the target. The load method depends on the LOAD_TYPE keyword. This option is not valid for Q subscriptions that specify stored procedures as targets.
- **E** Specifies a manual load. You can use your own load procedure or application to load the target table rather than using the Q Apply program. In this case, you use the LOADDONE command to indicate that the load is done.

**SUPPRESS DELETES**

Specifies whether to send rows that were deleted from the source table. This parameter is not valid for Classic sources.

- **N** Send deleted rows.
- **Y** Do not send deleted rows.

**CONFLICT ACTION**

Specifies what action to take if a conflict occurs.

- **I** Ignore.
- **F** Force. This action requires the send option CHANGED COLS ONLY = ‘N’.
- **D** Disable the Q subscription.
- **S** Stop Q Apply.
- **Q** Stop reading from the queue.

**ERROR ACTION**

Specifies what action to take if an error occurs.

- **S** Stop Q Apply without applying the transaction.
- **D** In Q replication, disable subscription and notify Q Capture. In Classic replication, disable subscription and notify the Classic capture components.
- **Q** Stop reading from the queue.

**OKSQLSTATES "sqlstates"**

Specifies a list of SQL statements within double quotation marks that are not to be considered as errors when applying changes to this table.

**LOAD TYPE**

Specifies a type of load.
Choose the best type automatically.

1 Use LOAD from CURSOR only. You cannot use this option with Classic sources.

2 Use EXPORT/IMPORT only. You cannot use this option with Classic sources.

3 Use EXPORT/LOAD only. You cannot use this option with Classic sources.

4 Load from Classic sources.

ADD COLS (trgcolname srccolname)
Specify to add one or more columns to the Q subscription. If trgcolname and srccolname are the same, only specify the trgcolname. You can use this parameter to add a new column to the subscription. You can also use the ALTER ADD COLUMN command to add the column if the column does not already exist in the target table.

This parameter is not valid for Classic sources.

Example 1
You cannot use this example with a Classic source. To alter a Q subscription for unidirectional replication and change the load type to an automatic load, send deleted rows, and stop reading from the queue if an error occurs:

```
ALTER QSUB EMPLOYEE0001 REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1
USING OPTIONS ALL CHANGED ROWS N HAS LOAD PHASE 1
SUPPRESS DELETES N CONFLICT ACTION F ERROR ACTION Q LOAD TYPE 1
```

Example 2
To alter a Q subscription for unidirectional replication by adding two columns that you want to begin replicating from the source table:

```
ALTER QSUB EMPLOYEE0001 REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1
USING OPTIONS ADD COLS (BONUS,COMM)
```

Example 3
To alter a Q subscription that has a CCD target so that it can manage a new SQL registration by creating this registration:

```
ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER CAPTURE TO QCAPDB;
SET SERVER TARGET TO QAPPDB;
SET CAPTURE SCHEMA SOURCE QCAP1;
SET APPLY SCHEMA QAPP1;
ALTER QSUB SUB1 REPLQMAP QCAPDB_QCAP1_TO_QAPPDB_QAPP1
MANAGE TARGET CCD CREATE SQL REGISTRATION;
```

Example 4
To alter a Q subscription that has a CCD target by deleting the SQL registration of its target CCD:

```
ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
```
CREATE QSUB command (unidirectional replication)

Use the CREATE QSUB command to create a Q subscription that maps a source table to a target table. For Classic replication, a Q subscription maps a source table or view in the Classic metadata catalog to a target table.

Syntax

```
CREATE QSUB
  SUBTYPE U  USING REPLQMAP mapname

  (SUBNAME subname
   DESC description
   REPLQMAP mapname
   src-clause
   trg-clause)

src-clause:

  source_name

    source_owner
    SRC OWNER LIKE "predicate1"
    SRC NAME LIKE "predicate2"
    SRC NAME LIKE "predicate"
    SRC ALL

  OPTIONS opt-clause

opt-clause:

  SEARCH CONDITION "search_condition"
  ALL CHANGED ROWS N Y

  HAS LOAD PHASE N
  SUPPRESS DELETES N Y

trg-clause:

  EXIST
  TARGET
    NAME target_name
    OWNER owner
    NAMING PREFIX prefix
```
TABLE OWNER: target_owner

TABLE NAME: target_name

NAMING PREFIX: prefix

SAME AS SOURCE

SAME AS USERID

FEDERATED: fed-clause

IN: tsname

DB: name

NAMING PREFIX: prefix

prof-clause

TYPE: USERTABLE

STOREDPROC

NICKNAME: targetcolumns

CCD

ccd-clause

MANAGE TARGET CCD: action

KEYS:

indexowner.indexname

NAMING PREFIX: prefix

NEW NICKNAME RMT SERVERNAME: srvname

owner.nickname

NAMING PREFIX: prefix

CONFLICT ACTION: IF DSQ

ERROR ACTION: S D Q

OKSQLSTATES: "sqlstates"

LOAD TYPE: 0 1 2 3 4

ZOS INDEX CREATE USING PROFILE: pname

NICKNAME: owner.nickname

NEW NICKNAME RMT SERVERNAME: srvname

owner.nickname

NAMING PREFIX: prefix

CONFLICT ACTION: IF DSQ

ERROR ACTION: S D Q

OKSQLSTATES: "sqlstates"

LOAD TYPE: 0 1 2 3 4

fed-clause:

nickname target name

ccd-clause:

CONDENSED ON OFF

COMPLETE ON OFF

WITH UOW COLS ALL

( colname )
prof-clause:

```
CREATE USING PROFILE pname
```

targetcolumns:

```
TRGCOLS ALL
INCLUDE (trgcolname srccolname)
EXCLUDE (trgcolname)
EXPRESSION (exp TARGET trgcolname)
srccolname
```

ccdoptions:

```
ALL,
COLS (AFTER aftcols BEFORE befoles)
OPTIONS BEFORE IMAGE COLUMNS
PREFIX "X"
INCLUDE (AFTER aftincludes BEFORE befoles)
FOR KEY COLS ONLY
```

action:

```
--CREATE SQL REGISTRATION
--ALTER SQL REGISTRATION FOR Q REPLICATION
```

Parameters

**SUBTYPE U**

Specifies unidirectional replication.

**USING REPLQMAP mapname**

Specifies the name of the replication queue map that is used by all of the Q subscriptions in this command. This is the replication queue map that will be used by all of the Q subscriptions in a mass scenario, or if replication queue maps are not specified with the parenthesis for each Q subscription.

**SUBNAME subname**

Specifies the name of the Q subscription.

**DESC "description"**

Specifies a description of the Q subscription.

**REPLQMAP mapname**

Specifies the name of the replication queue map for the Q subscription.

src-clause:
source_owner.source_name
   Specifies the source table’s schema and name.

SRC OWNER LIKE "predicate1"
   Specify to choose all tables with a schema that matches the expression in the LIKE statement. The following example shows a LIKE statement:
   CREATE QSUB USING REPLQMAP ABCDPUBQMAP
       (SRC OWNER LIKE "ASN%");
   CREATE QSUB USING REPLQMAP ABCDPUBQMAP
       (SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%);

SRC NAME LIKE
   Specify to choose all tables with a name that matches the expression in the LIKE statement. The following example shows a LIKE statement:
   CREATE QSUB USING REPLQMAP ABCDPUBQMAP
       (SRC OWNER LIKE "ASN%");
   CREATE QSUB USING REPLQMAP ABCDPUBQMAP
       (SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%);

SRC ALL
   Specify to choose all tables that exist on the source server. For DB2 sources, this excludes catalog views.

opt-clause:

SEARCH CONDITION "search_condition"
   Specifies a search condition for filtering changes to replicate. You cannot use this parameter with Classic replication. The change is not sent if the predicate is false. "search_condition" is an annotated select WHERE clause that must contain a colon before the column names of the table to be replicated. The following example shows a WHERE clause:
   CREATE QSUB USING REPLQMAP ASNMAP
       (SUBNAME mineid ALLTYPE1 OPTIONS SEARCH CONDITION
        "WHERE :MYKEY > 1000")

ALL CHANGED ROWS
   Specifies the data sending option.
   N   Send a row only if a subscribed column in the source table changes.
   Y   Send a row when any column in the source table changes.

HAS LOAD PHASE
   Specifies whether the target table for the Q subscription will be loaded with data from the source.
   N   No load phase at the target. This is the default.
   I   Specifies an automatic load. The Q Apply program loads the target. The load method depends on the LOAD_TYPE keyword. This parameter is not valid for Q subscriptions that specify stored procedures as targets.
   E   Specifies a manual load. An application other than the Q Apply program loads the target. In this case, you use the LOADDONE command to indicate that the load is done.

SPILL_MODELQ name
   Specifies the name of the model queue that is used as a spill queue for this Q subscription. On z/OS, you might want to create separate spill queues for Q subscriptions if the page set for the model queue is not large enough to handle transactions from multiple Q subscriptions during a load.
SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table. This parameter is not valid for Classic replication.
N Send deleted rows.
Y Do not send deleted rows.

trg-clause:
EXIST
Specifies that the target table exists.
• If you specify EXIST but do not provide a target table name, the ASNCLP program will look for the default table TGT-SOURCE TABLE NAME.
• If you specify EXIST and a single TARGET NAME, and you use SOURCE ALL or SOURCE NAME LIKE, then all of the source tables will be mapped to that single specified existing target table.
• If you do not specify EXIST, and you use SOURCE ALL or SOURCE NAME LIKE, then the source tables will be paired with target tables that use the default name TGT-SOURCE TABLE NAME.

TARGET
Specifies options for the target table owner and name.
NAME target_owner.target_name
Specifies the target table name and optionally the table schema.
NAMING PREFIX
Specifies the prefix to use to name the target table. The default is TGT. You can specify any other prefix, for example, if you specify CLP as a prefix and the source table is T1, the target table would be called CLPT1.

TABLE OWNER
Specifies options for the target table owner.
target_owner
Specifies to use the schema of the target table.
NAMING PREFIX
Specifies the prefix to use to name the target table owner. The default is TGT. You can specify any other prefix, for example, if you specify CLP as a prefix and the source table is T1, the target table would be called CLPT1.

SAME AS SOURCE
Specifies to use the same owner as the corresponding source table.
SAME AS USERID
Specifies to use the current user ID.

TABLE NAME
Specifies options for the target table name.
target_name
Specifies the name that you want to use for the target table.
NAMING PREFIX
Specifies the prefix to use to name the target table. For example, if you specify CLP as a prefix and the source table is T1, the target table would be called CLPT1.

SAME AS SOURCE
Specifies to name the target table the same as the corresponding source table.
IN

DB name
   Specifies the name of the logical database for the table space (required for z/OS).

tsname
   Specifies the name of the table space for the target table.

Federated targets: Specifies an existing table space (Oracle), segment (Sybase), dbspace (Informix), or file group (Microsoft SQL Server). This parameter is not applicable for Teradata targets.

NAMING PREFIX prefix
   Specifies the prefix to use to name the table space.

TYPE

USERTABLE
   Specifies a table as the target.

STOREDPROC
   Specifies a stored procedure as the target.

NICKNAME
   Specifies a nickname as the target.

CCD
   Specifies a consistent-change data (CCD) table as the target.

CREATE SQL REGISTRATION
   Registers the target CCD table for the Q subscription as a source for SQL replication.

ALTER SQL REGISTRATION FOR Q REPLICATION
   Modifies an existing registration for SQL replication by updating the CD_OWNER field in the IBMSNAP_REGISTER table with the Q Apply schema and the CD_TABLE field with the name of the receive queue for the Q subscription. You can also use this action to change an existing SQL registration to a Q subscription that uses a different receive queue.

KEYS
   Specifies one or more key columns that replication uses to determine the uniqueness of a row. If no key is specified, replication tries to determine its own key by looking first for a primary key within the set of replicated columns, then for a unique constraint, and then for a unique index. If none of these exists, replication will use all subscribed, valid columns as key columns for replication. (Some subscribed columns, such as LOB columns, cannot be used as keys.)

   indexowner,indexname
      Specifies the index owner and name.

   NAMING PREFIX prefix
      Specifies the prefix to use to name the index.

   keyname
      Specifies the name of the columns that are included in the index.

         + Ascending order.
         - Descending order.
**ZOS INDEX CREATE USING PROFILE** `pname`  
Specifies the name of the index profile for customizing a z/OS index.

**NICKNAME**  
Specifies the nickname for the Q Apply program to use to load rows into the target table with the LOAD from CURSOR utility.  

- `owner.nickname`  
  Specifies the source owner and nickname.

**NAMING PREFIX** `prefix`  
Specifies the prefix to use to name the nickname.

**NEW NICKNAME RMT SERVERNAME** `srvname`  
Specifies the name of the remote server if the ASNCLP program creates the nickname for loading.

**CONFLICT ACTION**  
Specifies what action to take if a conflict occurs.  

- `I` Ignore.  
- `F` Force: This action requires the send option `CHANGED COLS ONLY = 'N'`.  
- `D` Disable the Q subscription.  
- `S` Stop Q Apply.  
- `Q` Stop reading from queue.

**ERROR ACTION**  
Specifies what action to take if an error occurs.  

- `S` Stop Q Apply without applying the transaction.  
- `D` Disable the Q subscription and notify the Q Capture program or the Classic capture components.  
- `Q` Stop reading from the receive queue.

**OKSQLSTATES** `"sqlstates"`  
Specifies a list of SQL statements within double quotation marks that are not to be considered as errors when applying changes to this table.

**LOAD TYPE**  
Specifies a type of load.  

- `0` Choose the best type automatically.  
- `1` Use LOAD from CURSOR only. Not valid for Classic replication.  
- `2` Use SELECT/IMPORT only. Not valid for Classic replication.  
- `3` Use SELECT/LOAD only. Not valid for Classic replication.  
- `4` Load from Classic sources.

**fed-clause**

- `nickname target owner`  
  If you specify the FEDERATED keyword, you can optionally provide an owner for the nickname that is created for a federated target.

- `nickname target name`  
  If you specify the FEDERATED keyword, you can optionally provide a name for the nickname that is created for a federated target.
ccd-clause

**CONDENSED**

Specify one of the following values:

- **ON** Specifies that the CCD table is condensed. A condensed CCD table contains one row for every key value in the source table and contains only the latest value for the row.
- **OFF** Specifies that the CCD table is noncondensed. A noncondensed CCD table contains multiple rows with the same key value, one row for every change that occurs to the source table.

**COMPLETE**

Specify one of the following values:

- **ON** Specifies that the CCD table is complete. A complete CCD table contains every row of interest from the source table and is initialized with a full set of source data.
- **OFF** Specifies that the CCD table is noncomplete. A noncomplete CCD table contains only changes to the source table and starts with no data.

**WITH UOW COLS**

Specify one of the following values:

- **ALL** Specifies that the CCD table contains all four unit-of-work (UOW) columns: IBMSNAP_AUTHID, IBMSNAP_AUTHTKN, IBMSNAP_PLANID, IBMSNAP_UOWID.

**colname**

Specify one or more unit-of-work (UOW) columns for the CCD table.

targetcolums

**TRGCOLS**

- **ALL** Specify to replicate all columns from the source table.

**INCLUDE**

Specifies the column definitions if the target table does not exist. Specifies the replicated columns in the target table.

**trgcolname**

Specify to define a target table column that uses the provided name and the properties of a source column with the same name. In the following example, both the source and target table have the columns one, two, and three.

```
CREATE QSUB SUBTYPE U USING REPLQMAP replqmap9 (SUBNAME sprog64.srtable
EXIST TARGET NAME dprogpr64.trgtable
TRGCOLS INCLUDE (one, two))
```

**srccolname**

Specify to define a target table column that uses the provided name and to use the properties of the source column for the target column properties.

**EXCLUDE**

Specify to exclude the source column from the target table definition. This keyword can be used only when the source and target tables have the
same column names, or when you are creating a new target table. You cannot use this keyword when you are creating a new target table with a Classic replication source.

The following example shows how you would use the CREATE QSUB command with this option. In the example, the source table columns are one, two, and three.

CREATE QSUB USING REPLQMAP replqmap10
(SUBNAME sub10 dpropr64.srctable EXIST TARGET NAME
dpropr64.tgttable trgcols
EXCLUDE(three))

**EXPRESSION** exp
  Specifies a DB2-supported expression to which the target column is mapped.

**trgcolname**
  Specifies the name of the target column.

**ccdoptions**

**BEFORE IMAGE COLUMNS**
  Specifies that the before-image value of each added column will be replicated.

  **PREFIX "x"**
  Specifies the prefix for each before-image column. If you do not specify a prefix, the default, X, is used. If this prefix generates invalid names, then other letters will be used beginning with Y until valid names are generated.

  **ALL**
  Specifies that all of the after-image columns have before-image columns. This option is the default. Depending on the prefix that you choose, the DB2 database either picks before-image columns for existing targets or generates new before-image columns for new targets.

**COLS**
  Specifies custom before-image column names.

  **AFTER aftercols**
  Specifies the name of the after-image column in the target table.

  **BEFORE beforecols**
  Specifies the name of the before-image column in the target table. This parameter is mandatory. The value of **BEFORE** takes precedence over the name that is generated by the prefix for this particular column.

**INCLUDE**
  Specifies the columns that will be part of the before-image columns.

  **AFTER afterincludes**
  Specifies the name of the after-image column in the target table.

  **BEFORE beforeincludes**
  Specifies the name of the before-image column. This parameter is optional. The value of **BEFORE** takes precedence over the name that is generated by the prefix for this particular column.
FOR KEY COLS ONLY
   Specifies that before-image columns are generated only for the replication key columns.

prof-clause:
CREATE
   Specify to create a table space.

USING PROFILE pname
   Specifies the name of the profile to use to create the table space.

Usage notes
- The REPLQMAP keyword is mandatory. You can specify either CREATE QSUB USING REPLQMAP mapname or CREATE QSUB (SUBNAME subname REPLQMAP mapname).
- If a target table is specified and SRC ALL or SRC NAME LIKE was specified, all the source tables will attempt to subscribe to target tables with the same name.
- If the TABLE OWNER or TABLE NAME keywords are not specified, the default owner is the owner of the corresponding source table, and the default name is TGT-SOURCE TABLE NAME.
- The DB value for Logical Database is mandatory for target tables on z/OS products. It must be specified in the profile.
- If a mass subscription is used (for example, if you use the SRC OWNER LIKE or SRC NAME LIKE clause), the specified target_owner.target_name clause is valid only if the target table exists. Only the default or a naming prefix are allowed for generated target tables.
- The CREATE QSUB command performs an additional check when you create a Q subscription for a CCD target. If you configured Q Apply to manage an SQL Capture schema, and an SQL registration exists for the target CCD in this schema, the ASNCLP issues a message that Q Apply will manage the target CCD as an SQL replication source automatically.

Example 1

The following example shows the commands that are needed to set the environment and profiles for a CREATE QSUB command for unidirectional replication from a DB2 source. In this example, both the Q Capture program and Q Apply program run in the same z/OS subsystem and share a queue manager.

ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE to dbALIAS ECO6V71A DBNAME stlec1 ID ADMF001 password "xx";
SET SERVER TARGET to dbALIAS ECO6V71A DBNAME stlec1 ID ADMF001 password "xxx";
SET CAPTURE SCHEMA SOURCE QDECODER;
SET APPLY SCHEMA QDECODER;
SET QMANAGER "CSQ1" FOR CAPTURE SCHEMA);
SET QMANAGER "CSQ1" FOR APPLY SCHEMA;
SET PROFILE "UITRGT" FOR OBJECTS TARGET INDEX OPTIONS ZOS
   BUFFERPOOL BP1 STOGROUP "DPROSTG"
   PRIQTY ABSOLUTE 100 SECQTY ABSOLUTE 50;
SET PROFILE "UITRGT" FOR OBJECT TARGET TABLESPACE OPTIONS ZOS
   DB "JUTRDB"
   BUFFERPOOL BP4
   ENCODING UNICODE
   STOGROUP "DPROSTG"
   PRIQTY ABSOLUTE 100 SECQTY ABSOLUTE 50;
SET OUTPUT CAPTURE SCRIPT "capfile6.sql" TARGET SCRIPT "tgtfile.sql";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
Example 2

This example creates a Q subscription SUB_T1 that specifies an automatic load (LOAD TYPE 1) and creates a new nickname REPLDBA.NICK_T1 at the Q Apply server for the LOAD from CURSOR utility. RMTSAMPLE is the remote server definition on TESTDB that points to the SAMPLE database, which is the data source for the nickname.

ASNCMP SESSION SET TO Q REPLICAATION;
SET OUTPUT CAPTURE SCRIPT "REPLCAP.SQL" TARGET SCRIPT "REPLAPP.SQL";
SET LOG "QSUB.LOG";
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB TESTDB;
SET APPLY SCHEMA ASN;
SET CAPTURE SCHEMA SOURCE ASN;
CREATE QSUB (SUBNAME "SUB_T1" REPLQMAP SAMPLE ASN TO TESTDB ASN REPLDBA.T_TEMP
OPTIONS HAS LOAD PHASE I TARGET NAME REPLDBA.T_TEMPNEWNEW TYPE USERTABLE
NEW NICKNAME RMT SERVERNAME RMTSAMPLE REPLDBA.NICK_T1 LOAD TYPE 1);

Example 3

This examples creates the SUB_T2 Q subscription and specifies that the Q Apply program use an existing nickname, REPLDBA.NICK_T2, for the LOAD from CURSOR utility.

ASNCMP SESSION SET TO Q REPLICAATION;
SET OUTPUT CAPTURE SCRIPT "REPLCAP.SQL" TARGET SCRIPT "REPLAPP.SQL";
SET LOG "QSUB.LOG";
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB TESTDB;
SET APPLY SCHEMA ASN;
SET CAPTURE SCHEMA SOURCE ASN;
CREATE QSUB (SUBNAME "SUB_T2" REPLQMAP SAMPLE ASN TO TESTDB ASN REPLDBA.T_TEMP
OPTIONS HAS LOAD PHASE I TARGET NAME REPLDBA.T_TEMPNEWNEW TYPE USERTABLE
NEW NICKNAME REPLDBA.NICK_T2 LOAD TYPE 1);

Example 4

This example demonstrates the use of a naming prefix for the target table (XNEW) and tablespace for the target table (Y). The example also shows the use of "like" statements to specify the source table for the Q subscription.

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE
"%EMP%" TARGET TABLE NAME NAMING PREFIX XNEW IN DB D1CDG01 NAMING PREFIX Y);

Example 5

This example shows how to use a tablespace profile (USING PROFILE UTRGTS) for the target table tablespace when the target tables do not exist.

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE
"%EMP%" TARGET TABLE NAME NAMING PREFIX XNEW2 IN DB D1CDG01 EMPTBSP2 CREATE USING
PROFILE UTRGTS);

Example 6

This example shows that no IN clause is required when the target table exists.

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE
"%EMP%" EXIST TARGET TABLE OWNER NAMING PREFIX X);

Example 7

This example creates all of the target tables in one tablespace (RST1).
CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE "%EMP%" TARGET TABLE NAME XNEW IN DB D1CDG01 RTS1);

**Example 8**

In this example, the target table exists, the target owner is ABC, and target table prefix is XNEW.

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE "%EMP%" TARGET TABLE OWNER ABC TABLE NAME NAMING PREFIX XNEW );

**Example 9**

This example shows the use of a target owner prefix (ABC).

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE "%EMP%" TARGET TABLE OWNER NAMING PREFIX ABC TABLE NAME NAMING PREFIX XNEW );

**Example 10**

In this example the source and target owner names are the same. For the source and target owner names to be the same, the target must be in a different database or subsystem than the source.

CREATE QSUB USING REPLQMAP QDECODERQM (SRC OWNER LIKE "DSN8710%" SRC NAME LIKE "%EMP%" TARGET TABLE OWNER SAME AS SOURCE TABLE NAME SAME AS SOURCE );

**Example 11**

This example does not use the environment and profile from [Example 1 on page 98](#). It creates a Q subscription for unidirectional replication from a DB2 source using the replication queue map SAMPLE_ASN1_TO_TARGETDB_ASN1 and specifies that the Q Apply program loads the target tables using the EXPORT and IMPORT utilities. It also specifies that the column EMPNO be used as the key for replication.

CREATE QSUB USING REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1 (SUBNAME EMPLOYEE0001 EMPLOYEE OPTIONS HAS LOAD PHASE I TARGET NAME TGTEMPLOYEE KEYS (EMPNO) LOAD TYPE 2);

**Example 12**

This example creates a Q subscription from the DB2 table EMPLOYEE to the Sybase table TGT_EMPLOYEE. The table will be created in the existing Sybase segment SEG_EMPLOYEE by using the SAMPLE_ASN_TO_FEDDB_ASN replication queue map. The table will have the nickname of EMPNICKNAME.

CREATE QSUB USING REPLQMAP SAMPLE_ASN_TO_FEDDB_ASN (SUBNAME FEDQSUB EMPLOYEE TARGET NAME TGTEMPLOYEE FEDERATED EMPNICKNAME);

**Example 13**

This example creates a Q subscription with a new target CCD table. All of the columns in the source table are in the Q subscription and all of the columns in the target will have before-image columns.

ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB SAMPLE;
SET CAPTURE SCHEMA SOURCE ASNAPP1;
SET APPLY SCHEMA ASNAPP1;
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1_TO_SAMPLE ASNAPP1
(SUBNAME TESTCCCDNEW DATA.EMPLOYEE TARGET NAME DATA.TGTEMPLOYEE
TYPE CCD CONDENSED ON COMPLETE ON WITH UOW COLS ALL
TRGCOLS ALL BEFORE IMAGE COLUMNS ALL);

Example 14

This example creates a Q subscription with new target CCD table. All of the
columns in the source table take part in the Q subscription. The command also
specifies before-image columns for the key columns and a before image prefix

ASNCLP SESSION SET TO Q REPLICA;  
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;  
SET SERVER CAPTURE TO DB SAMPLE;  
SET SERVER TARGET TO DB SAMPLE;  
SET CAPTURE SCHEMA SOURCE ASNAPP1;  
SET APPLY SCHEMA ASNAPP1;  
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1_TO_SAMPLE ASNAPP1
(SUBNAME TESTCCCDNEW DATA.EMPLOYEE TARGET NAME DATA.TGTEMPLOYEE
TYPE CCD CONDENSED ON COMPLETE ON WITH UOW COLS ALL
TRGCOLS ALL BEFORE IMAGE COLUMNS PREFIX Y FOR KEYS COLS ONLY);

Example 15

This example creates a Q subscription with a new CCD target table. All of the
columns in the source table are in the Q subscription. The command specifies a
subset of columns that will have before images. The command also specifies the
before-image column names for these columns.

ASNCLP SESSION SET TO Q REPLICA;  
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;  
SET SERVER CAPTURE TO DB SAMPLE;  
SET SERVER TARGET TO DB SAMPLE;  
SET CAPTURE SCHEMA SOURCE ASNAPP1;  
SET APPLY SCHEMA ASNAPP1;  
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1_TO_SAMPLE ASNAPP1
(SUBNAME TESTCCCDNEW DATA.EMPLOYEE TARGET NAME DATA.TGTEMPLOYEE
TYPE CCD CONDENSED ON COMPLETE ON WITH UOW COLS ALL
TRGCOLS ALL BEFORE IMAGE COLUMNS INCLUDE
(AFTER C1 BEFORE BEFC1, AFTER C2 BEFORE BEFC2, AFTER C3 BEFORE BEFC3);

Example 16

This example creates a Q subscription with a new target CCD table. The before
image columns exist for all the replicated columns in the target. Some columns
have a before-image prefix of Y while the others have no specific prefix.

ASNCLP SESSION SET TO Q REPLICA;  
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;  
SET SERVER CAPTURE TO DB SAMPLE;  
SET SERVER TARGET TO DB SAMPLE;  
SET CAPTURE SCHEMA SOURCE ASNAPP1;  
SET APPLY SCHEMA ASNAPP1;  
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1_TO_SAMPLE ASNAPP1
(SUBNAME TESTCCCDEXIST DATA.EMPLOYEE TARGET EXIST NAME DATA.TGTEMPLOYEE
TYPE CCD CONDENSED ON COMPLETE ON WITH UOW COLS ALL
TRGCOLS ALL BEFORE IMAGE COLUMNS PREFIX Y ALL COLS
(AFTER C1 BEFORE BEFC1, AFTER C2 BEFORE BEFC2);

Example 17

This example creates a Q subscription with a new target CCD table. Only a subset
of the columns in the target table participate in replication, and before image
columns exist only for three columns in the target table. The before-image columns
do not have a specific prefix and have different names for each after-image
column.

ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB SAMPLE;
SET CAPTURE SCHEMA SOURCE ASNAPP1;
SET APPLY SCHEMA ASNAPP1;
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1 TO SAMPLE ASNAPP1
  (SUBNAME TESTCCDOEXIST DATA.EMPLOYEE TARGET EXIST NAME DATA.TGTEMPLOYEE
  TYPE CCD CONDENSED ON COMPLETE ON WITH UOW COLS ALL
  TRGCOLS INCLUDE (C1, C2, C3, C4, C5) BEFORE IMAGE COLUMNS INCLUDE
  (AFTER C1 BEFORE BEFC1, AFTER C2 BEFORE BEFC2, AFTER C3 BEFORE BEFC3);

Example 18

This example creates a Q subscription by using a target column expression that
maps all of the columns that match the expression CONCAT(:C1,:C2) to the target
column CEXP.

ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER CAPTURE TO DB SAMPLE;
SET SERVER TARGET TO DB SAMPLE;
SET CAPTURE SCHEMA SOURCE ASNAPP1;
SET APPLY SCHEMA ASNAPP1;
CREATE QSUB USING REPLQMAP SAMPLE ASNAPP1 TO SAMPLE ASNAPP1
  (SUBNAME TESTEXPRESSTION DATA.EMPLOYEE TARGET NAME DATA.TGTEMPLOYEE
  TRGCOLS ALL EXPRESSION ("CONCAT(:C1,:C2)" TARGET CEXP));

Example 19

This example creates a Q subscription called CLASSIC0001 for Classic replication. The CREATE QSUB command specifies a source table called CLASSICTABLE and specifies that the Q Apply program is to load a target table of the same name.

ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE "asnservers.ini"
  ID CLASSICADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET APPLY SCHEMA ASN1;
CREATE QSUB USING REPLQMAP CLASSIC ASN1 TO TARGET ASN1 (SUBNAME CLASSIC0001
  CLASSICTABLE OPTIONS HAS LOAD PHASE I TARGET NAME CLASSICTABLE LOAD TYPE 4);

**DROP QSUB command (unidirectional Q replication)**

Use the DROP QSUB command to delete a Q subscription for unidirectional Q replication.

**Syntax**

```sql
DROP QSUB ALL [USING REPLQMAP–mapname]
```
### Parameters

**ALL**
Specify to delete all Q subscriptions. If you specify this parameter, you cannot combine it with any other parameters.

**USING REPLQMAP** *mapname*
Specify to delete all of the Q subscriptions that use the specified replication queue map.

**SUBNAME** *subname*
Specifies the name of the Q subscription to delete.

**USING REPLQMAP** *mapname*
Specifies the name of the replication queue map that is used by the Q subscription that you want to delete.

**FOR SUBNAME LIKE** "*predicate*"
Specify to delete all of the Q subscriptions that match the expression in the LIKE statement. The following example shows a LIKE statement:

```
DROP QSUB USING REPLQMAP ABCDREPLQMAP
  (FOR SUBNAME LIKE "ASN%");
```

### Example 1
To delete a Q subscription for unidirectional replication from a relational source:

```
DROP QSUB (SUBNAME EMPLOYEE0001 USING REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1)
```

### Example 2
To delete a Q subscription for unidirectional replication from a non-relational source:

```
DROP QSUB (SUBNAME EMPLOYEE0001 USING REPLQMAP CLASSIC_ASN_TO_TARGETDB_ASN1)
```

### LIST QSUB command (Q replication)
Use the LIST command to list Q subscriptions.

**Syntax**

```
LIST QSUB
  FOR TABLEOWNER—ownername
  FOR QMAP—mapname
  FOR QCAPTURE
  FOR QAPPLY
  SCHEMA—schema
  SERVER—dbparms
```

dbparms-clause:
Parameters

FOR TABLEOWNER ownername
List only the Q subscriptions dedicated to the specified table owner name.

FOR QMAP mapname
List only the Q subscriptions used by the specified replication queue map.

QCAPTURE
List the Q subscription information that is defined in a single set of Q Capture control tables. Use this parameter with the CONFIG SERVER parameter to specify a Classic source.

QAPPLY
List the Q subscription information that is defined in a single set of Q Apply control tables.

SCHEMA schema
Specifies which schema to use. The default is "ASN".

dbparms-clause:

DB dbalias
Specifies the database alias name.

DBALIAS aliasname
Specifies the database alias name.

DBNAME dbname
Specifies the database name.

ID userid
Specifies the user ID to use to connect to the database.

PASSWORD pwd
Specifies the password to use for connections.

CONFIG SERVER servername

Classic sources: Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP should use to connect to the Classic server.

FILE filename
Specifies the complete path and file name to the Classic replication configuration file. If you do not use the FILE parameter, the ASNCLP attempts to use the asnservers.ini file in the current directory, if that file exists.

Example 1

This example lists the Q subscriptions with Q Capture schema ASN. (The SET SERVER command determines which database or subsystem the Q Capture schema is located on.)

LIST QSUB FOR QCAPTURE SCHEMA ASN;
Example 2

This example lists the Q subscriptions on server CLASSIC1 with schema ASN.
LIST QSUB FOR QCAPTURE SCHEMA ASN CONFIG SERVER CLASSIC1 FILE asnservers.ini ID id1 PASSWORD "passwd1";

LIST REPLQMAP command (Q replication)
Use the LIST REPLQMAP command to list replication queue maps.

Syntax

```
LIST REPLQMAP FOR QCAPTURE QAPPLY SCHEMA schema
```

```
SERVER dbparms
```

dbparms-clause:

```
DB dbalias
DBALIAS aliasname
DBNAME dbname
ID userid
PASSWORD pwd
CONFIG SERVER servername
FILE filename
```

Parameters

QCAPTURE
List the replication queue map information that is defined in a single set of Q Capture control tables. Use this parameter with the CONFIG SERVER parameter to specify a Classic source.

QAPPLY
List the replication queue map information that is defined in a single set of Q Apply control tables.

SCHEMA schema
Specifies which schema to use. The default is "ASN".

dbparms-clause:

DB dbalias
Specifies the database alias name.

Linux UNIX Windows DBALIAS aliasname
Specifies the database alias name.

z/OS DBNAME dbname
Specifies the database name.

ID userid
Specifies the user ID to use to connect to the database.

PASSWORD pwd
Specifies the password to use for connections.

CONFIG SERVER servername
**Classic sources**: Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP should use to connect to the Classic server.

**FILE filename**

Specifies the complete path and file name to the Classic replication configuration file. If you do not use the **FILE** parameter, the ASNCLP attempts to use the `asnservers.ini` file in the current directory, if that file exists.

**Example 1**

This example lists the replication queue maps with Q Capture schema ASN. (The SET SERVER command determines which database or subsystem the Q Capture schema is located on.)

```
LIST REPLQMAP FOR QCAPTURE SCHEMA ASN;
```

**Example 2**

This example lists the replication queue maps on server CLASSIC1 with schema ASN.

```
LIST REPLQMAP FOR QCAPTURE SCHEMA ASN CONFIG SERVER CLASSIC1 FILE asnservers.ini ID id1 PASSWORD "passwd1";
```

**SET DROP command (unidirectional replication)**

Use the SET DROP command to specify whether to drop the target table and its table space when you delete a Q subscription for unidirectional replication. You also use this command to specify whether to drop the table spaces for control tables.

**Syntax**

```
SET DROP TARGET NEVER
```

```
SET DROP CONTROL TABLES WHEN EMPTY NEVER
```

**Parameters**

**TARGET**

Specifies if you want to drop the target tables with the subscription.

- **ALWAYS**
  
  Always drop the target table.

- **NEVER**
  
  Never drop the target table.

**DROP**

Specify what you want to drop when you delete a Q subscription.

**TARGET**

- Target table.

**CONTROL TABLES**

- Q Capture and Q Apply control tables.
TABLESPACE
Specifies whether the table space should be dropped when the target table or control tables that it contains is dropped.

WHEN EMPTY
   Drop the table space only when it is empty.

NEVER
   Never drop the table space.

Example 1
To always drop the target table when the Q subscription is deleted:
SET DROP TARGET ALWAYS

Example 2
To never drop the table space for the control tables when the control tables are dropped.
SET DROP CONTROL TABLES TABLESPACE NEVER

ASNCLP commands for multidirectional Q replication

The ASNCLP commands for multidirectional replication define, change, and drop the objects that are unique to bidirectional and peer-to-peer Q replication.

The following topics demonstrate how you can combine multidirectional Q replication commands to create ASNCLP setup scripts:

- “Sample ASNCLP scripts for setting up bidirectional Q replication” on page 63
- “Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers)” on page 68
- “Sample ASNCLP scripts for setting up peer-to-peer Q replication (three servers)” on page 73

Table 9 lists the ASNCLP commands for event publishing and links to topics that describe each command.

Table 9. ASNCLP commands for multidirectional Q replication

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change a Q subscription for bidirectional replication</td>
<td>&quot;ALTER QSUB command (bidirectional replication)&quot; on page 109</td>
</tr>
<tr>
<td>Change a Q subscription for peer-to-peer replication</td>
<td>&quot;ALTER QSUB command (peer-to-peer replication)&quot; on page 111</td>
</tr>
<tr>
<td>Change a replication queue map</td>
<td>ALTER REPLQMAP command</td>
</tr>
<tr>
<td>Establish a session for Q replication</td>
<td>ASNCLP SESSION SET TO command</td>
</tr>
<tr>
<td>Create the control tables for the Q Capture and Q Apply programs</td>
<td>CREATE CONTROL TABLES FOR command</td>
</tr>
<tr>
<td>Create a Q subscription for bidirectional replication</td>
<td>&quot;CREATE QSUB command (bidirectional replication)&quot; on page 113</td>
</tr>
<tr>
<td>Create a Q subscription for peer-to-peer replication</td>
<td>&quot;CREATE QSUB command (peer-to-peer replication)&quot; on page 116</td>
</tr>
<tr>
<td>Create a replication queue map</td>
<td>CREATE REPLQMAP command</td>
</tr>
<tr>
<td>Description</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Drop the control tables for the Q Capture and Q Apply programs</td>
<td>DROP CONTROL TABLES ON command</td>
</tr>
<tr>
<td>Delete a replication queue map</td>
<td>DROP REPLQMAP command</td>
</tr>
<tr>
<td>Delete the subgroup that you set by using the SET SUBGROUP command.</td>
<td>&quot;DROP SUBGROUP command (multidirectional Q replication)&quot; on page 118</td>
</tr>
<tr>
<td>Delete a Q subscription for bidirectional replication</td>
<td>&quot;DROP SUBTYPE command (bidirectional replication)&quot; on page 118</td>
</tr>
<tr>
<td>Delete a Q subscription for peer-to-peer replication between two servers</td>
<td>&quot;DROP SUBTYPE command (peer-to-peer replication)&quot; on page 119</td>
</tr>
<tr>
<td>List Q Apply schemas</td>
<td>LIST APPLY SCHEMA command</td>
</tr>
<tr>
<td>List Q Capture schemas</td>
<td>LIST CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Signal that a manual load of the target table is complete</td>
<td>LOAD DONE command</td>
</tr>
<tr>
<td>Invoke ASNCLP program scripts used to set up multidirectional replication</td>
<td>&quot;LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication)&quot; on page 120</td>
</tr>
<tr>
<td>Set the Q Apply schema for all task commands</td>
<td>SET APPLY SCHEMA command</td>
</tr>
<tr>
<td>Set the Q Capture schema for all task commands</td>
<td>SET CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Connect the servers that are used for bidirectional or peer-to-peer replication.</td>
<td>&quot;SET CONNECTION command (multidirectional Q replication)&quot; on page 122</td>
</tr>
<tr>
<td>Specify whether the ASNCLP will enforce matching constraints between the source and target tables.</td>
<td>&quot;SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication)&quot; on page 122</td>
</tr>
<tr>
<td>Define the log file for the ASNCLP program</td>
<td>SET LOG command</td>
</tr>
<tr>
<td>Specify the Q Capture and Q Apply schema on a server that is used for multidirectional replication</td>
<td>&quot;SET MULTIDIR SCHEMA command (multidirectional Q replication)&quot; on page 124</td>
</tr>
<tr>
<td>Define output files that contain SQL scripts for multidirectional replication</td>
<td>&quot;SET OUTPUT command (multidirectional Q replication)&quot; on page 125</td>
</tr>
<tr>
<td>Specify custom parameters for database objects to be created implicitly</td>
<td>SET PROFILE command</td>
</tr>
<tr>
<td>Set the WebSphere MQ queue manager</td>
<td>SET QMANAGER command</td>
</tr>
<tr>
<td>Set a reference table to identify a Q subscription that you want to change or delete.</td>
<td>&quot;SET REFERENCE TABLE command (multidirectional Q replication)&quot; on page 127</td>
</tr>
<tr>
<td>Specify the server that contains both Q Capture and Q Apply control tables to use in the ASNCLP session</td>
<td>&quot;SET SERVER command (multidirectional Q replication)&quot; on page 128</td>
</tr>
<tr>
<td>Specify the name of the subgroup, a collection of Q subscriptions between servers that are used for multidirectional replication</td>
<td>&quot;SET SUBGROUP command (multidirectional Q replication)&quot; on page 129</td>
</tr>
<tr>
<td>Specify the tables that participate in a bidirectional or peer-to-peer configuration</td>
<td>&quot;SET TABLES command (multidirectional Q replication)&quot; on page 130</td>
</tr>
<tr>
<td>Enable and disable the trace for the ASNCLP commands</td>
<td>SET TRACE command</td>
</tr>
<tr>
<td>Display the environment set during the session</td>
<td>SHOW SET ENV command</td>
</tr>
<tr>
<td>Start a Q subscription</td>
<td>START QSUB command</td>
</tr>
<tr>
<td>Stop a Q subscription</td>
<td>STOP QSUB command</td>
</tr>
<tr>
<td>Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions.</td>
<td>VALIDATE WSMQ ENVIRONMENT FOR command</td>
</tr>
</tbody>
</table>
Table 9. ASNCLP commands for multidirectional Q replication (continued)

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map.</td>
<td>VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command</td>
</tr>
</tbody>
</table>

**ALTER QSUB command (bidirectional replication)**

Use the ALTER QSUB command to change the properties of one or both bidirectional Q subscriptions for a single logical table.

**Syntax**

```
ALTER QSUB SUBTYPE B

FROM NODE—servername.schemaname
    SOURCE—source-clause
    TARGET—target-clause

FROM NODE—servername.schemaname
    SOURCE—source-clause
    TARGET—target-clause
```

**source-clause:**

```
<table>
<thead>
<tr>
<th>ALL CHANGED ROWS</th>
<th>N</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAS LOAD PHASE</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>
```

**target-clause:**

```
| CONFLICT RULE    | K | C | A |
| CONFLICT ACTION  | I | F | D | S |
| ERROR ACTION     | Q | D | S |

LOAD TYPE 0 2 3 OKSQLSTATES—"sqlstates" |
```

**Parameters**

**SUBTYPE B**

Specifies bidirectional Q subscriptions.

**FROM NODE server.schemaname**

Identifies one of the two bidirectional Q subscriptions by specifying the server and schema of its source table.

**source-clause:**

**ALL CHANGED ROWS**

Specifies the data sending option.

N  Send a row only if a subscribed column in the source table changes.
Y  Send a row when any column in the source table changes.

HAS LOAD PHASE
Specifies whether the target table for the Q subscription will be loaded with data from the source.
N  No load phase at the target. This is the default.
I  Specifies an automatic load. The Q Apply program calls the EXPORT and IMPORT utilities or EXPORT and LOAD utilities, depending on the type of load that is specified in the LOAD_TYPE keyword and on the platform of the Q Apply server and Q Capture server.
E  Specifies a manual load. An application other than the Q Apply program loads the target table. In this case, you insert the LOADDONE signal (using the LOADDONE command) into the IBMQREP_SIGNAL table at the Q Capture server to inform the Q Capture program that the application is done loading.

target-clause:

CONFLICT RULE
K  Check only key values.
C  Check changed non-key values in addition to key values.
A  Check all values for updates.

CONFLICT ACTION
Specifies what action to take if a conflict occurs.
I  Ignore.
F  The Q Apply program tries to force the change. This requires that the Q Capture program send all columns, so the CHANGED_COLS_ONLY value must be set to N (no) in the IBMQREP_SUBS table.
D  Disable the Q subscription.
S  Stop Q Apply.
Q  Stop reading from queue.

ERROR ACTION
Specifies what action to take if an error occurs.
S  Stop Q Apply without applying the transaction.
D  Disable the Q subscription and notify Q Capture.
Q  Stop reading from queue.

OKSQLSTATES "sqlstates"
Specifies a list of SQL statements within double quotation marks that are not to be considered as errors when applying changes to this table.

LOAD TYPE
Specifies a type of load.
0  Choose the best type automatically.
2  Use EXPORT and IMPORT only.
3  Use EXPORT and LOAD only.
Usage notes

You must use the LOAD MULTIDIR REPL SCRIPT command to invoke the script file that contains this command and the other commands that are required to set its context.

Example

The following script changes the Q subscriptions for the EMPLOYEE table at SAMPLE and SAMPLE2. For the Q subscription whose source table is at SAMPLE (FROM NODE SAMPLE.RED), the load option will be changed to manual load. For the other Q subscription, the error action is changed to disable the Q subscription and notify the Q Capture program if an error occurs.

To identify the Q subscriptions, the first commands identify the subgroup, the servers in the subgroup, and the reference table RED.EMPLOYEE.

```
SET SUBGROUP "BIDIRGROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET REFERENCE TABLE USING SCHEMA "SAMPLE".RED USES TABLE RED.EMPLOYEE;
```

```
ALTER QSUB SUBTYPE B
FROM NODE SAMPLE.RED SOURCE HAS LOAD PHASE E
FROM NODE SAMPLE2.BLUE TARGET ERROR ACTION D;
```

**ALTER QSUB command (peer-to-peer replication)**

Use the ALTER QSUB command to change the properties of the peer-to-peer Q subscriptions for a single logical table.

**Syntax**

```
   >>>ALTER QSUB---SUBTYPE---P---SOURCE source-clause [TARGET target-clause]
```

source-clause:

```
   [HAS LOAD PHASE] [N I E]
```

target-clause:

```
   [ERROR ACTION] [O S D]
   [LOAD TYPE] [0 2 3]
   [OKSQLSTATES] "sqlstates"
```

**Parameters**

**SUBTYPE P**

Specifies a peer-to-peer Q subscription.
source-clause:

HAS LOAD PHASE
Specifies whether the target table for the Q subscription will be loaded with data from the source.

N No load phase at the target. This is the default.
I Specifies an automatic load. The Q Apply program calls the EXPORT and IMPORT utilities or EXPORT and LOAD utilities, depending on the type of load that is specified in the LOAD_TYPE keyword, and on the platform of the Q Apply server and Q Capture server.
E Specifies a manual load. An application other than the Q Apply program loads the target table. In this case, you insert the LOADDONE signal (using the LOADDONE command) into the IBMQREP_SIGNAL table at the Q Capture server to inform the Q Capture program that the application is done loading.

target-clause:

ERROR ACTION
D Disable subscription and notify the Q Capture program.
S Stop the Q Apply program without applying the transaction.
Q Stop reading from the receive queue.

LOAD TYPE
Specifies a type of load.
0 Choose the best type automatically.
2 Use EXPORT and IMPORT only.
3 Use EXPORT and LOAD only.

OKSQLSTATES "sqlstates"
Specifies a list of SQL statements within double quotation marks that are not to be considered as error when applying changes to this table.

Usage notes
You must use the LOAD MULTIDIR REPL SCRIPT command to invoke the script file that contains this command and the other commands that are required to set its context.

Example
The following script changes the Q subscriptions for the STAFF table at SAMPLE, SAMPLE2, and SAMPLE3 in a peer-to-peer configuration with three servers. The command specifies an automatic load that uses the EXPORT and IMPORT utilities and sets the error action to disable the Q subscription and notify the Q Capture program if an error occurs.

To identify the Q subscriptions, the first commands identify the subgroup, the servers in the subgroup, and the reference table GRAY.STAFF.

```
SET SUBGROUP "P2P3GROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET SERVER MULTIDIR TO DB "SAMPLE3";
```
SET REFERENCE TABLE USING SCHEMA "SAMPLE".GRAY USES TABLE GRAY.STAFF;

ALTER QSUB SUBTYPE P SOURCE HAS LOAD PHASE I TARGET ERROR ACTION D LOAD TYPE 2;

**CREATE QSUB command (bidirectional replication)**
Use the CREATE QSUB command to create two Q subscriptions for a single logical table that participates in bidirectional replication.

**Syntax**

```
CREATE QSUB SUBTYPE B

FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause

FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause
```

**source-clause:**

- ALL_CHANGED_ROWS=N
- HAS_LOAD_PHASE=I

**target-clause:**

- CONFLICT_RULE=K
- CONFLICT_ACTION=I
- ERROR_ACTION=D
- LOAD_TYPE=0
- OKSQLSTATES="sqlstates"

**Parameters**

**SUBTYPE B**
Specifies bidirectional Q subscriptions.

**FROM NODE** servername.schemaname
A FROM NODE statement is required if you want to specify options for one or both of the Q subscriptions. If you omit FROM NODE, both Q subscriptions will be created with the following default options:
- ALL_CHANGED_ROWS=N
- BEFORE_VALUES=N
- CHANGED_COLS_ONLY=Y
- HAS_LOADPHASE=I
- CONFLICT_ACTION=K
- CONFLICT_RULE=I
• ERROR_ACTION=Q

In the FROM NODE statement, you specify a server name and schema to identify the logical table that is the source for the Q subscription.

source-clause:

ALL CHANGED ROWS
Specifies the data sending option.

N Send a row only if a subscribed column in the source table changes.

Y Send a row when any column in the source table changes.

HAS LOAD PHASE
Specifies whether the target table for the Q subscription will be loaded with data from the source.

N No load phase at the target. This is the default.

I Specifies an automatic load. The Q Apply program calls the EXPORT and IMPORT utilities or EXPORT and LOAD utilities, depending on the type of load that is specified in the LOAD_TYPE keyword and on the platform of the Q Apply server and Q Capture server.

E Specifies a manual load. An application other than the Q Apply program loads the target table. In this case, you insert the LOADDONE signal (using the LOADDONE command) into the IBMQREP_SIGNAL table at the Q Capture server to inform the Q Capture program that the application is done loading.

target-clause:

CONFLICT RULE

K Check only key values.

C Check changed non-key values in addition to key values.

A Check all values for updates.

CONFLICT ACTION

I Ignore.

F The Q Apply program tries to force the change. This requires that the Q Capture program send all columns, so the CHANGED_COLS_ONLY value must be set to N (no) in the IBMQREP_SUBS table.

D Disable the Q subscription.

S Stop the Q Apply program.

Q Stop reading from the receive queue.

ERROR ACTION
Specifies what action to take if an error occurs.

Q Stop reading from the receive queue.

D Disable the Q subscription and notify the Q Capture program.

S Stop the Q Apply program without applying the transaction.

OKSQLSTATEs "sqlstates"
Specifies a list of SQL statements within double quotation marks that are not to be considered as errors when applying changes to this table.
LOAD TYPE
Specifies the utilities that the Q Apply program uses to load the target.

- 0  Choose the best type automatically.
- 2  Use EXPORT and IMPORT only.
- 3  Use EXPORT and LOAD only.

Usage notes

[Table 10] shows the permitted combinations for BEFORE_VALUES and CHANGE_COLS_ONLY depending on the values of CONFLICT_RULE and CONFLICT_ACTION.

Recommendation: Always use the ASNCLP or Replication Center to change the value of CONFLICT_RULE and CONFLICT_ACTION. The administration tools will automatically set the correct value for BEFORE_VALUES and CHANGE_COLS_ONLY. Neither of these attributes can be set explicitly using the administration tools.

Table 10. Required attributes for BEFORE_VALUES and CHANGE_COLS_ONLY depending on the values of CONFLICT_RULE and CONFLICT_ACTION

<table>
<thead>
<tr>
<th>CONFLICT RULE</th>
<th>CONFLICT ACTION</th>
<th>BEFORE_VALUES</th>
<th>CHANGE COLS ONLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>I, S, D, or Q</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>K</td>
<td>F</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td>I, S, D, or Q</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>A</td>
<td>I, S, D, or Q</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

Example

The following commands create two Q subscriptions for bidirectional replication between the SAMPLE and SAMPLE2 servers. The commands specify an automatic load at both servers. At SAMPLE, a CONFLICT_RULE of C (check changed key and non-key values) and a CONFLICT_ACTION of F (force the change) are specified. At SAMPLE2, a CONFLICT_RULE of A (check all values for updates) and a CONFLICT_ACTION of I (ignore) are specified.

To identify the Q subscriptions, the first commands identify the subgroup, the servers and schemas in the subgroup, and the two replication queue maps. The SET TABLES command specifies the RED.EMPLOYEE table at the SAMPLE database, which will generate statements to create a matching table at SAMPLE2.

```sql
SET SUBGROUP "bidirgroup"
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET MULTIDIR SCHEMA "SAMPLE".RED;
SET MULTIDIR SCHEMA "SAMPLE2".BLUE;
SET CONNECTION SOURCE "SAMPLE".RED TARGET "SAMPLE2".BLUE REPLQMAP "SAMPLE_RED_TO_SAMPLE2_BLUE";
SET CONNECTION SOURCE "SAMPLE2".BLUE TARGET "SAMPLE".RED REPLQMAP "SAMPLE2_BLUE_TO_SAMPLE_RED";
```
CREATE QSUB command (peer-to-peer replication)

Use the CREATE QSUB command to create a set of Q subscriptions for a single logical table that participates in peer-to-peer replication.

Syntax

```plaintext
CREATE QSUB SUBTYPE P [SOURCE source-clause] [TARGET target-clause]
```

source-clause:

- `HAS LOAD PHASE`:
  - `N` No load phase. This is the default.
  - `I` Specifies an automatic load. The Q Apply program calls the EXPORT and IMPORT utilities or EXPORT and LOAD utilities, depending on the type of load that is specified in the LOAD TYPE keyword, and on the platform of the Q Apply server and Q Capture server.
  - `E` Specifies a manual load. An application other than the Q Apply program loads the target table. In this case, you insert the LOADDONE signal (using the LOADDONE command) into the IBMQREP_SIGNAL table at the Q Capture server to inform the Q Capture program that the application is done loading.

target-clause:

- `ERROR ACTION`:
  - `0` Specifies the LOAD type as 0.
  - `2` Specifies the LOAD type as 2.
  - `3` Specifies the LOAD type as 3.
  - `0` QSQLSTATES="sqlstates"
**ERROR ACTION**
Specifies what action to take if an error occurs.
- **Q** Stop reading from the receive queue.
- **D** Disable subscription and notify the Q Capture program.
- **S** Stop the Q Apply program without applying the transaction.

**LOAD TYPE**
Specifies a type of load.
- **0** Choose the best type automatically.
- **2** Use EXPORT and IMPORT only.
- **3** Use EXPORT and LOAD only.

**OKSQLSTATES "sqlstates"**
Specifies a list of SQL statements within double quotation marks that are not to be considered as error when applying changes to this table.

**Usage notes**
- Convergence columns and triggers will be created on the tables that participate in the peer-to-peer replication setup.
- For peer-to-peer replication with convergence, only the attributes shown in Table 11 are allowed (and are implicitly assigned).

Table 11. Attributes for peer-to-peer replication with convergence

<table>
<thead>
<tr>
<th>Conflict Rule</th>
<th>Conflict Action</th>
<th>Before Values</th>
<th>Change Cols Only</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>F</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

**Example**

The following script creates Q subscriptions for the STAFF table at SAMPLE, SAMPLE2, and SAMPLE3 in a peer-to-peer configuration with three servers. The Q subscriptions specify no load phase and an error action that prompts the Q Apply program to stop reading from the receive queue if an error occurs.

To identify the Q subscriptions, the first commands identify the subgroup, the servers and schemas in the subgroup, and the replication queue maps. The SET TABLES command specifies GRAY.STAFF at the SAMPLE database, which will generate SQL statements to create matching tables at the other two servers.

```
SET SUBGROUP "p2p3group";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET SERVER MULTIDIR TO DB "SAMPLE3";
SET MULTIDIR SCHEMA "SAMPLE".GRAY;
SET MULTIDIR SCHEMA "SAMPLE2".BROWN;
SET MULTIDIR SCHEMA "SAMPLE3".YELLOW;
```

```
SET CONNECTION SOURCE "SAMPLE".GRAY TARGET "SAMPLE2".BROWN REPLQMAP "SAMPLE_GRAY_TO_SAMPLE2_BROWN";
SET CONNECTION SOURCE "SAMPLE".GRAY TARGET "SAMPLE3".YELLOW REPLQMAP "SAMPLE_GRAY_TO_SAMPLE3_YELLOW";
SET CONNECTION SOURCE "SAMPLE2".BROWN TARGET "SAMPLE".GRAY REPLQMAP "SAMPLE2_BROWN_TO_SAMPLE_GRAY";
SET CONNECTION SOURCE "SAMPLE2".BROWN TARGET "SAMPLE3".YELLOW REPLQMAP "SAMPLE2_BROWN_TO_SAMPLE3_YELLOW";
```
SET CONNECTION SOURCE "SAMPLE3".YELLOW TARGET "SAMPLE".GRAY REPLQMAP "SAMPLE3_YELLOW_TO_SAMPLE_GRAY";
SET CONNECTION SOURCE "SAMPLE3".YELLOW TARGET "SAMPLE2".BROWN REPLQMAP "SAMPLE3_YELLOW_TO_SAMPLE2_BROWN";
SET TABLES (SAMPLE.GRAY.GRAY.STAFF);
CREATE QSUB SUBTYPE P SOURCE HAS LOAD PHASE N TARGET ERROR ACTION Q;

DROP SUBGROUP command (multidirectional Q replication)
Use the DROP SUBGROUP command to delete the subgroup that you set by using the SET SUBGROUP command.

Syntax

DROP SUBGROUP

Usage notes
When you delete a subgroup, all Q subscriptions within the group are also deleted.

Example 1
The following script drops the bidirectional subgroup BIDIGROUP. First it sets the subgroup, then sets the two servers in the group. The SET MULTIDIR SCHEMA command specifies the shared Q Capture and Q Apply schema RED at one of the servers to further identify the Q subscriptions that are dropped at both servers along with the subgroup.

SET SUBGROUP "BIDIRGROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE1";
SET MULTIDIR SCHEMA "SAMPLE".RED
DROP SUBGROUP;

DROP SUBTYPE command (bidirectional replication)
Use the DROP SUBTYPE command to delete both bidirectional Q subscriptions for a single logical table.

The command creates SQL statements to connect to both servers in the bidirectional configuration and delete the Q subscription from their control tables.

Syntax

DROP SUBTYPE B QSUBS

Parameters
B Specifies bidirectional Q replication.
QSUBS Specifies that all of the Q subscriptions that are defined with the same SET SUBGROUP command will be deleted.
Usage notes

- No tables or table spaces are ever dropped.
- You must use the LOAD MULTIDIR REPL SCRIPT command to invoke the script file that contains this command and the other commands that are required to set its context.

Example

The following commands delete the Q subscription for the EMPLOYEE table at SAMPLE and SAMPLE2. To identify the Q subscription, the first commands identify the subgroup, the servers in the subgroup, and the reference table RED.EMPLOYEE.

```
SET SUBGROUP "BIDIRGROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET REFERENCE TABLE USING SCHEMA "SAMPLE".RED USES TABLE RED.EMPLOYEE;
DROP SUBTYPE B QSUBS;
```

DROP SUBTYPE command (peer-to-peer replication)

Use the DROP SUBTYPE command to delete the peer-to-peer Q subscriptions for a single logical table.

The command creates SQL statements to connect to all servers in the peer-to-peer configuration and delete the Q subscription from their control tables.

Syntax

```
DROP SUBTYPE P QSUBS
```

Parameters

**SUBTYPE P**

Specifies a peer-to-peer Q subscription.

Usage notes

- No tables or table spaces are ever dropped.
- Convergence columns and triggers will remain on the tables that previously participated in a peer-to-peer replication scenario.
- You must use the LOAD MULTIDIR REPL SCRIPT command to invoke the script file that contains this command and the other commands that are required to set its context.

Example

The following script deletes the Q subscription for the STAFF table at SAMPLE, SAMPLE2, and SAMPLE3. To identify the Q subscription, the first commands identify the subgroup, the servers in the subgroup, and the reference table GRAY.STAFF.

```
SET SUBGROUP "P2P3GROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE2";
```
SET SERVER MULTIDIR TO DB "SAMPLE3";
SET REFERENCE TABLE USING SCHEMA "SAMPLE.GRAY" USES TABLE GRAY.STAFF;
DROP SUBTYPE P QSUBS;

LOAD MULTIDIR RECP SCRIPT command (multidirectional Q replication)

Use the LOAD MULTIDIR RECP SCRIPT command to invoke ASNCLP program scripts used to set up peer-to-peer and bidirectional replication.

Syntax

```
LOAD MULTIDIR RECP SCRIPT "filelocation/filename"
```

Parameters

**filelocation**

Specifies the absolute path where the input file is located. If no directory is specified, the current directory is assumed.

**filename**

Specifies the name of the bidirectional or peer-to-peer replication input file.

Usage notes

- Only definitions pertaining to one subgroup can be placed in one bidirectional or peer-to-peer replication script.
- Several scripts can be invoked to set up several subgroups if each one is invoked with its own LOAD MULTIDIR RECP SCRIPT call.
- Several LOAD MULTIDIR RECP SCRIPT statements can exist in one ASNCLP program input file.

Example

The following is a sample script used to invoke four bidirectional or peer-to-peer scripts:

```
LOAD MULTIDIR RECP SCRIPT "3nodes\3Node0.in";
LOAD MULTIDIR RECP SCRIPT "3nodes\3Node1.in";
LOAD MULTIDIR RECP SCRIPT "3nodes\3Node2.in";
LOAD MULTIDIR RECP SCRIPT "3nodes\3Node3.in";
```

**Note:** This script creates four subgroups. Each subgroup definition is placed into a bidirectional or peer-to-peer script (for example, 3Node0.in).

The following is a sample bidirectional or peer-to-peer script (3Node0.in):

```
# Give the subgroup a name.
set subgroup "3Node0";

# Set the servers (databases) that will participate in this subgroup.
set server multidir to db "testdb";
set server multidir to db "testdb1";
set server multidir to db "testdb2";

# Specify the Q Capture/Q Apply schema for the catalogs used on those servers.
set multidir schema "testdb".BLUE;
set multidir schema "testdb1".RED;
```
set multidir schema "testdb2".YELLOW;

# Specify the replication queue maps used to join the catalogs together
set connection SOURCE "testdb".BLUE TARGET "testdb1".RED replqmap "BLUEtoRED";
set connection SOURCE "testdb".BLUE TARGET "testdb".YELLOW replqmap "BLUEtoYELLOW";
set connection SOURCE "testdb1".RED TARGET "testdb".BLUE replqmap "REDtoBLUE";
set connection SOURCE "testdb2".YELLOW TARGET "testdb".BLUE replqmap "YELLOWtoBLUE";
set connection SOURCE "testdb1".RED TARGET "testdb2".YELLOW replqmap "REDtoYELLOW";
set connection SOURCE "testdb2".YELLOW TARGET "testdb".BLUE replqmap "YELLOWtoRED";

# Specify the tables to participate in this subgroup (1 per server).
set tables("testdb".BLUE.BLUE.AllTypes0, "testdb1".RED.RED.AllTypes0, "testdb2".YELLOW.YELLOW.AllTypes0);
# Create the subgroup
create qsub subtype p;

This bidirectional or peer-to-peer script creates a subgroup “3Node0”. All of the information required to generate the subgroup’s Q subscriptions is located in this one input file.

**SET BIDI NODE command**

Use the SET BIDI NODE command to specify the paired Q Capture and Q Apply control tables that will be the source of replication definitions to promote to another server in a bidirectional configuration.

### Syntax

```
SET BIDI NODE number SERVER DBALIAS dbalias DBNAME dbname
```

### promote-options:

```
PROMOTE TO promote-svr-options SCHEMA promschema
```

### promote-svr-options:

```
DBALIAS dbalias DBNAME dbname ID userid PASSWORD pwd
```

### Parameters

**NODE number**

Specifies node 1 or 2 of the bidirectional configuration. A node represents a combination of server and schema.

**SERVER**

Specifies the source database alias name. This is the database that contains the configuration that is being promoted.

**DBALIAS dbalias**

Specifies the database alias name.
Specifies the database name.

ID userid
Specifies the user ID to use when you connect to the source database.

PASSWORD pwd
Specifies the password to use when you connect to the source server that is specified by the SERVER parameter. If you specify the user ID and do not specify the password, you will be prompted to enter the password. The password is hidden as you type.

SCHEMA schema
Specifies the schema that contains the configurations to promote from the source server. The source server is specified by the SERVER DBALIAS or DBNAME parameters.

To specify the nodes to promote configurations from and the corresponding destination nodes:

SET BIDI NODE 1 SERVER DBALIAS TEST01 ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS PRODUCTION01 ID id1 PASSWORD "pw1d" SCHEMA ASN;

SET BIDI NODE 2 SERVER DBALIAS TEST02 ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS TEST011 ID id1 PASSWORD "pw1d" SCHEMA ASN;

**Example**

Use the SET CONNECTION command to connect the two servers that are used for bidirectional or peer-to-peer replication.
Syntax

```plaintext
SET CONNECTION SUBNAME subscriptionname SOURCE sourceservername.sourceschemaname
TARGET targetservername.targetschemaname REPLQMAP mapname
```

Parameters

**SUBNAME** *subscriptionname*
Specifies the name of the Q subscription between the two servers (from source to target) that are specified in the connection. If more than one Q subscription is created between the two servers, the first Q subscription will carry the name as specified, and every subsequent Q subscription will have an incremental number appended to it.

**SOURCE**

*sourceservername*
Specifies the name of the source server.

*sourceschemaname*
Specifies the schema of the control tables at the source server.

**TARGET**

*targetservername*
Specifies the name of the target server.

*targetschemaname*
Specifies the schema of the control tables at the target server.

**REPLQMAP** *mapname*
Specifies the name of the replication queue map that connects the Q Capture program at the source server with the Q Apply program at the target server.

Usage notes

To make a connection between two servers, you must run the SET CONNECTION command twice because both servers act as a source and a target. See the example below.

Example

To set the connection between the servers BLUE and RED that are used for peer-to-peer replication servers:
```
SET CONNECTION SOURCE TESTDB.BLUE
TARGET TESTDB1.RED REPLQMAP BLUE.TO.RED;
```
```
SET CONNECTION SOURCE TESTDB1.RED
TARGET TESTDB.BLUE REPLQMAP RED.TO.BLUE
```

**SET ENFORCE MATCHING CONSTRAINTS command**

*(multidirectional Q replication)*

Use the SET ENFORCE MATCHING CONSTRAINTS command to specify whether the ASNCLP will enforce matching constraints between the source and target tables. The ASNCLP by default checks that referential integrity constraints, check constraints, and unique constraints match for the source and target tables.
Syntax

```
SET ENFORCE MATCHING CONSTRAINTS [YES | NO]
```

Parameters

**YES**

Specify to enforce referential integrity constraints, check constraints, and unique constraints.

**NO**

Specify to not enforce matching constraints on source and target tables.

Usage notes

When you specify **NO**, you can subscribe a child table before subscribing the parent table.

Example

```
SET SUBGROUP "P2PSUBGROUP";
SET SERVER MULTIDIR TO DB SAMPLE;
SET SERVER MULTIDIR TO DB TEMPDB;
SET MULTIDIR SCHEMA SAMPLE.ASN;
SET MULTIDIR SCHEMA TEMPDB.ASN;
SET CONNECTION SOURCE SAMPLE.ASN TARGET TEMPDB.ASN REPLQMAP "RQ1"
SET CONNECTION SOURCE TEMPDB.ASN TARGET SAMPLE.ASN REPLQMAP "RQ2";
SET ENFORCE MATCHING CONSTRAINTS NO;
SET TABLES (SAMPLE.ASN.DB2OWNER.TEMP_FK, TEMPDB.ASN.DB2OWNER.TEMP_FK);
CREATE QSUB SUBTYPE P;
```

**SET MULTIDIR SCHEMA command (multidirectional Q replication)**

Use the **SET MULTIDIR SCHEMA** command to set the same schema for the Q Capture and Q Apply control tables on a server that is used for bidirectional or peer-to-peer replication. If you do not specify the server or schema, the ASNCLP program defaults to ASN.

Syntax

```
SET MULTIDIR SCHEMA servername.schemaname
```

Parameters

**servername**

Specifies the name of the server that contains the Q Capture and Q Apply control tables.

**schemaname**

Specifies the schema for the Q Capture and Q Apply control tables on a server that is used for bidirectional or peer-to-peer replication.

Example

To set the multidirectional schema for bidirectional or peer-to-peer replication to **BLUE** on the server **TESTDB**:
SET MULTIDIR SCHEMA TESTDB.BLUE

SET OUTPUT command (multidirectional Q replication)

Use the SET OUTPUT command to define output files for the ASNCLP program. The output files contain the SQL statements needed to set up multidirectional Q replication, or the ASNCLP commands needed to promote a replication environment.

Syntax

```
SET OUTPUT MULTIDIR MONITOR SCRIPT "monfname" PROMOTE SCRIPT "profname"
```

Parameters

MULTIDIR
- Specify to name the output files after the databases that the SQL scripts run on.

MONITOR SCRIPT "monfname"
- Specifies the output file name for scripts that run at the Monitor control server. The default file name is replmonitor.sql.

PROMOTE SCRIPT "profname"
- Specifies the output file name for the ASNCLP commands generated by PROMOTE statements. If the file name is not specified, the default file created is named qrepl_asnclp.in.

Usage notes

- If a script already exists, the new script appends to the current script.
- MULTIDIR does not require a file name because the ASNCLP program automatically names the output SQL scripts based on the names of the databases that the SQL scripts run on.
- The double quotation marks in the command syntax are required.

Example 1

To name the SQL script output files based on the names of the databases that the SQL script runs on:

```
SET OUTPUT MULTIDIR
```

SET PEER NODE command

Use the SET PEER NODE command to specify the paired Q Capture and Q Apply control tables that will be the source of replication definitions to promote to another server in a peer-to-peer configuration.

Syntax

```
SET PEER NODE number SERVER dbalias DBNAME dbname
```

Chapter 3. ASNCLP commands for Q replication and Classic replication
promote-options:

-PROMOTE TO- promote-svr-options | SCHEMA promschema |

promote-svr-options:

-DBALIAS dbalias | DBNAME dbname | ID userid | PASSWORD pwd |

Parameters

**NODE number**
Specify with a digit from 1 to 6 a node in the peer-to-peer configuration that defines the overall peer-to-peer context to be promoted. A node represents a set of Q Capture and Q Apply programs that are on the same server, have the same schema, and are involved in the peer-to-peer configuration. Up to six nodes can be identified in a peer-to-peer configuration, each defined with a separate SET PEER NODE command.

**SERVER**
Specifies the source database alias name. This is the database that contains the configuration that is being promoted.

**DBALIAS aliasname**
Specifies the database alias name.

-DBNAME dbname
Specifies the database or subsystem name.

**ID userid**
Specifies the user ID to use when you connect to the source database.

**PASSWORD pwd**
Specifies the password to use when you connect to the source server. If you specify the user ID and do not specify the password, you will be prompted to enter the password. The password is hidden as you type.

**SCHEMA schema**
Specifies the source schema name.

promote-options

-PROMOTE TO- promote-svr-options | SCHEMA promschema |

**SCHEMA promschema**
Specifies the schema of the control tables in the destination database. If the schema is not specified, then the schema in the source configuration is used in the generated scripts for the promoted configuration.
DBALIAS aliasname
Specifies the destination database alias name.

DBNAME dbname
Specifies the destination database name.

ID userid
Specifies the user ID of the destination database for promotion. The resulting promotion commands will not include a user ID if this parameter is not specified.

PASSWORD pwd
Specifies the password to use to connect to the destination database. The resulting promotion commands will not include a password if this parameter is not specified.

Example
To specify the nodes to promote configurations from and the corresponding destination nodes:

SET PEER NODE 1 SERVER DBALIAS AMERICAS ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS AMERICAS01 ID id1 PASSWORD "pw1d" SCHEMA ASN;

SET PEER NODE 2 SERVER DBALIAS EUROPE ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS EUROPE01 ID id1 PASSWORD "pw1d" SCHEMA ASN;

SET PEER NODE 3 SERVER DBALIAS ASIA ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS ASIA01 ID id1 PASSWORD "pw1d" SCHEMA ASN;

SET REFERENCE TABLE command (multidirectional Q replication)
Use the SET REFERENCE TABLE command to identify a Q subscription for bidirectional or peer-to-peer replication. You specify this command before you use the ALTER QSUB or DROP SUBTYPE commands to change or drop the Q subscriptions.

Syntax

```
SET REFERENCE TABLE— USING SCHEMA—server.schema—USES TABLE—tableowner.tablename
```

Parameters

**USING SCHEMA**

server
Specifies the name of the server that contains the table.

schema
Specifies the schema of the control tables in which this table is specified as a source and target.

**USES TABLE**

tableowner
Specifies the table schema.

tablename
Specifies the table name.
Example 1

The following script sets the reference table RED.DEPARTMENT at the server SAMPLE to identify and change the Q subscription for the DEPARTMENT table at SAMPLE and SAMPLE1.

```
SET SUBGROUP "BIDIRGROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE1";
SET REFERENCE TABLE USING SCHEMA "SAMPLE".RED USES TABLE RED.DEPARTMENT;
ALTER QSUB SUBTYPE B SOURCE HAS LOAD PHASE I TARGET ERROR ACTION S;
```

Example 2

The following script sets the reference table RED.EMPLOYEE at the server SAMPLE to identify and drop the Q subscription for the EMPLOYEE table at SAMPLE, SAMPLE1, and SAMPLE2.

```
SET SUBGROUP "P2P3GROUP";
SET SERVER MULTIDIR TO DB "SAMPLE";
SET SERVER MULTIDIR TO DB "SAMPLE1";
SET SERVER MULTIDIR TO DB "SAMPLE2";
SET REFERENCE TABLE USING SCHEMA "SAMPLE".RED USES TABLE RED.EMPLOYEE;
DROP SUBTYPE P QSUBS;
```

SET SERVER command (multidirectional Q replication)

Use the SET SERVER command to specify the server that contains both Q Capture and Q Apply control tables (MULTIDIR) to use in the ASNCLP session. After you set a server name, all subsequent commands in the session will apply to this server until you change the server with this command.

Syntax

```plaintext
SET SERVER CAPTURE TARGET MULTIDIR TO NULLS DBALIAS aliasname other-options
```

other-options:

```
ID userid
PASSWORD pwd
```

Parameters

**CAPTURE**

Specify to set the database as a Q Capture server.

**TARGET**

Specify to set the database as a Q Apply server (also referred to as target server).
MULTIDIR
Specify to set the database as a bidirectional or peer-to-peer replication server.
For z/OS, this is the subsystem location name.

NULLS
Specify to set the server name to NULL. This option resets a previously set
server name.

DBALIAS aliasname
Specifies the database alias name.

ID userid
Specifies the user ID to use to connect to the database.

PASSWORD pwd
Specifies the password to use to connect to the database. If you specify the
user ID and do not specify the password, you will be prompted to enter the
password.

Usage notes
When using bidirectional or peer-to-peer replication, you must explicitly set the
MULTIDIR option.

Example - basic
To set the bidirectional or peer-to-peer replication server to the database TESTDB:
SET SERVER MULTIDIR TO DBALIAS TESTDB

Example - password prompting
To set the bidirectional or peer-to-peer replication server and specify only the user
ID in the command:
SET SERVER MULTIDIR TO DBALIAS SAMPLE ID DB2ADMIN
You are prompted to enter the password. If you are running the commands from
an input file in batch mode, the program waits for you to enter the password
before the program processes the next commands.

SET SUBGROUP command (multidirectional Q replication)
Use the SET SUBGROUP command to specify a name for a collection of Q
subscriptions that are involved in bidirectional or peer-to-peer replication.

Syntax

```plaintext
SET SUBGROUP—subgroup-name
```

Parameters

`subgroupname`
Specifies the name of the collection of Q subscriptions for bidirectional or
peer-to-peer replication.
Example

To set the subgroup BLUEandRED:

SET SUBGROUP BLUEandRED

SET TABLES command (multidirectional Q replication)

Use the SET TABLES command to specify the tables that participate in a single bidirectional or peer-to-peer subscription (each listed table is both a source and a target for the Q subscription).

Syntax

```
SET TABLES

(server.schema.table_owner.table_name, server.schema.table_owner.table_name)
```

Parameters

- **server**
  - Specifies the name of the server (database) that contains the table.
- **schema**
  - Specifies the schema of the control tables in which this table is specified as a source or target.
- **tableowner**
  - Specifies the schema of the table.
- **tablename**
  - Specifies the name of the table.

Usage notes

- You must specify at least one table.
  - The first table must be located at the starting peer (peer-to-peer replication) or primary server (bidirectional replication), and it must already exist.
  - If you specify additional tables that already exist at the other servers, the ASNCLP program will check to see if they exist. If the tables do not exist, they will be created based on the first table.
- You must specify a CREATE QSUB command after identifying the tables for the Q subscription with the SET TABLES command.
- To create a set of Q subscriptions for peer-to-peer or bidirectional replication using the tables specified in the SET TABLES command, you must issue a CREATE QSUB command before the next SET TABLES command. That is, each SET TABLES command will override the previous one until you issue a CREATE QSUB statement.

Example 1

In this example, the table specified in parentheses is BLUE.TABLE3 on the testdb server with a Q Capture and Q Apply schema of BLUE. There are two other servers in the peer-to-peer configuration: testdb1 with a shared schema of RED and
testdb2 with a shared schema of GREEN. New tables will be generated on testdb1
and testdb2 with the names RED.TGTTABLE3 and GREEN.TGTTABLE3 because
no tables were specified explicitly for the RED and GREEN servers.

SET TABLES ("testdb",BLUE,BLUE,TABLE3);
CREATE QSUB SUBTYPE P;

Example 2

In this example, the first table specified in the SET TABLES command is
RCTEST2.TABLE2 on the testdb server with a Q Capture and Q Apply schema of
BLUE. New tables will be generated on testdb1 and testdb2 with the name of
RCTEST3.XYZ and RCBLUE.AllTypes0 because the two other tables are specified
explicitly.

SET TABLES ("testdb",BLUE,RTEST2.TABLE2,"testdb1",RED.RTEST3.XYZ,
"testdb2",YELLOW.RCBLUE.AllTypes0);
CREATE QSUB SUBTYPE P;

ASNCLP commands for unidirectional Q replication

The ASNCLP commands for unidirectional Q replication set the environment,
define, change, and delete Q subscriptions, and specify output files. Some of the
ASNCLP commands for unidirectional replication also apply to Classic replication.

“Sample ASNCLP scripts for setting up unidirectional Q replication” on page 57
and “Sample ASNCLP scripts for setting up unidirectional Q replication from a
Classic data source” on page 61 demonstrate how you can combine ASNCLP
commands to create an ASNCLP setup script.

Table 8 on page 82 lists the ASNCLP commands for unidirectional Q replication
and links to topics that describe each command.

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a column to a Q subscription</td>
<td>ALTER ADD COLUMN command</td>
</tr>
<tr>
<td>Update the IBMQREP_CAPPARMS table when you replicate from a Classic source</td>
<td>“ALTER CAPTURE PARAMETERS command (Classic replication)” on page 84</td>
</tr>
<tr>
<td>Change a Q subscription</td>
<td>“ALTER QSUB command (unidirectional replication)” on page 89</td>
</tr>
<tr>
<td>Change a replication queue map</td>
<td>ALTER REPLQMAP command</td>
</tr>
<tr>
<td>Establish a session for Q replication</td>
<td>ASNCLP SESSION SET TO command</td>
</tr>
<tr>
<td>Create a Q subscription</td>
<td>“CREATE QSUB command (unidirectional replication)” on page 89</td>
</tr>
<tr>
<td>Create the control tables for the Q Capture and Q Apply programs</td>
<td>CREATE CONTROL TABLES FOR command</td>
</tr>
<tr>
<td>Create a replication queue map</td>
<td>CREATE REPLQMAP command</td>
</tr>
<tr>
<td>Delete a Q subscription</td>
<td>“DROP QSUB command (unidirectional Q replication)” on page 102</td>
</tr>
<tr>
<td>Drop the control tables for the Q Capture and Q Apply programs</td>
<td>DROP CONTROL TABLES ON command</td>
</tr>
<tr>
<td>Delete a replication queue map</td>
<td>DROP REPLQMAP command</td>
</tr>
<tr>
<td>List Q subscriptions</td>
<td>“LIST QSUB command (Q replication)” on page 103</td>
</tr>
<tr>
<td>List replication queue maps</td>
<td>“LIST REPLQMAP command (Q replication)” on page 105</td>
</tr>
</tbody>
</table>
Table 12. ASNCLP commands for unidirectional Q replication (continued)

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>List Q Apply schemas</td>
<td>LIST APPLY SCHEMA command</td>
</tr>
<tr>
<td>List Q Capture schemas</td>
<td>LIST CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Signal that a manual load of the target table is complete</td>
<td>LOAD DONE command</td>
</tr>
<tr>
<td>Promote a Q subscription</td>
<td>PROMOTE QSUB command</td>
</tr>
<tr>
<td>Promote a replication queue map</td>
<td>PROMOTE REPLQMAP command</td>
</tr>
<tr>
<td>• Specify whether to drop the target table when you delete a Q subscription</td>
<td></td>
</tr>
<tr>
<td>• Specify whether to drop the table space when you drop the target table or control tables</td>
<td>“SET DROP command (unidirectional replication)” on page 106</td>
</tr>
<tr>
<td>Set the Q Apply schema for all task commands</td>
<td>SET APPLY SCHEMA command</td>
</tr>
<tr>
<td>Set the Q Capture schema for all task commands</td>
<td>SET CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Define the log file for the ASNCLP program</td>
<td>SET LOG command</td>
</tr>
<tr>
<td>Define output files that contain SQL statements to set up unidirectional Q replication</td>
<td>SET OUTPUT command</td>
</tr>
<tr>
<td>Specify custom parameters for database objects to be created implicitly</td>
<td>SET PROFILE command</td>
</tr>
<tr>
<td>Set the WebSphere MQ queue manager</td>
<td>SET QMANAGER command</td>
</tr>
<tr>
<td>Specify whether to automatically run each task command from an input file before the ASNCLP program processes the next task command</td>
<td>SET RUN SCRIPT command</td>
</tr>
<tr>
<td>Specify the Q Capture server or Q Apply server to use in the ASNCLP session for unidirectional replication.</td>
<td>SET SERVER command</td>
</tr>
<tr>
<td>Enable and disable the trace for the ASNCLP commands</td>
<td>SET TRACE command</td>
</tr>
<tr>
<td>Display the environment set during the session</td>
<td>SHOW SET ENV command</td>
</tr>
<tr>
<td>Start a Q subscription</td>
<td>START QSUB command</td>
</tr>
<tr>
<td>Stop a Q subscription</td>
<td>STOP QSUB command</td>
</tr>
<tr>
<td>Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions</td>
<td>VALIDATE WSMQ ENVIRONMENT FOR command</td>
</tr>
<tr>
<td>Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map</td>
<td>VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command</td>
</tr>
</tbody>
</table>

**ASNCLP commands for multidirectional Q replication**

The ASNCLP commands for multidirectional replication define, change, and drop the objects that are unique to bidirectional and peer-to-peer Q replication.

The following topics demonstrate how you can combine multidirectional Q replication commands to create ASNCLP setup scripts:

- “Sample ASNCLP scripts for setting up bidirectional Q replication” on page 63
- “Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers)” on page 68
Table 9 on page 107 lists the ASNCLP commands for event publishing and links to topics that describe each command.

**Table 13. ASNCLP commands for multidirectional Q replication**

<table>
<thead>
<tr>
<th>Description</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change a Q subscription for bidirectional replication</td>
<td>ALTER QSUB command (bidirectional replication) on page 107</td>
</tr>
<tr>
<td>Change a Q subscription for peer-to-peer replication</td>
<td>ALTER QSUB command (peer-to-peer replication) on page 111</td>
</tr>
<tr>
<td>Change a replication queue map</td>
<td>ALTER REPLQMAP command</td>
</tr>
<tr>
<td>Establish a session for Q replication</td>
<td>ASNCLP SESSION SET TO command</td>
</tr>
<tr>
<td>Create the control tables for the Q Capture and Q Apply programs</td>
<td>CREATE CONTROL TABLES FOR command</td>
</tr>
<tr>
<td>Create a Q subscription for bidirectional replication</td>
<td>CREATE QSUB command (bidirectional replication) on page 112</td>
</tr>
<tr>
<td>Create a Q subscription for peer-to-peer replication</td>
<td>CREATE QSUB command (peer-to-peer replication) on page 116</td>
</tr>
<tr>
<td>Create a replication queue map</td>
<td>CREATE REPLQMAP command</td>
</tr>
<tr>
<td>Drop the control tables for the Q Capture and Q Apply programs</td>
<td>DROP CONTROL TABLES ON command</td>
</tr>
<tr>
<td>Delete a replication queue map</td>
<td>DROP REPLQMAP command</td>
</tr>
<tr>
<td>Delete the subgroup that you set by using the SET SUBGROUP command.</td>
<td>DROP SUBGROUP command (multidirectional Q replication) on page 118</td>
</tr>
<tr>
<td>Delete a Q subscription for bidirectional replication</td>
<td>DROP SUBTYPE command (bidirectional replication) on page 118</td>
</tr>
<tr>
<td>Delete a Q subscription for peer-to-peer replication between two servers</td>
<td>DROP SUBTYPE command (peer-to-peer replication) on page 119</td>
</tr>
<tr>
<td>List Q Apply schemas</td>
<td>LIST APPLY SCHEMA command</td>
</tr>
<tr>
<td>List Q Capture schemas</td>
<td>LIST CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Signal that a manual load of the target table is complete</td>
<td>LOAD DONE command</td>
</tr>
<tr>
<td>Invoke ASNCLP program scripts used to set up multidirectional replication</td>
<td>LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication) on page 120</td>
</tr>
<tr>
<td>Set the Q Apply schema for all task commands</td>
<td>SET APPLY SCHEMA command</td>
</tr>
<tr>
<td>Set the Q Capture schema for all task commands</td>
<td>SET CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Connect the servers that are used for bidirectional or peer-to-peer replication.</td>
<td>SET CONNECTION command (bidirectional Q replication) on page 122</td>
</tr>
<tr>
<td>Specify whether the ASNCLP will enforce matching constraints between the source and target tables.</td>
<td>SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication) on page 123</td>
</tr>
<tr>
<td>Define the log file for the ASNCLP program</td>
<td>SET LOG command</td>
</tr>
<tr>
<td>Specify the Q Capture and Q Apply schema on a server that is used for multidirectional replication.</td>
<td>SET MULTIDIR SCHEMA command (multidirectional Q replication) on page 124</td>
</tr>
<tr>
<td>Define output files that contain SQL scripts for multidirectional replication</td>
<td>SET OUTPUT command (multidirectional Q replication) on page 125</td>
</tr>
<tr>
<td>Specify custom parameters for database objects to be created implicitly</td>
<td>SET PROFILE command</td>
</tr>
<tr>
<td>Description</td>
<td>Command</td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Set the WebSphere MQ queue manager</td>
<td>SET QMANAGER command</td>
</tr>
<tr>
<td>Set a reference table to identify a Q subscription that you want to change or delete.</td>
<td>&quot;SET REFERENCE TABLE command (multidirectional Q replication)&quot; on page 127</td>
</tr>
<tr>
<td>Specify the server that contains both Q Capture and Q Apply control tables to use in the ASNCLP session</td>
<td>&quot;SET SERVER command (multidirectional Q replication)&quot; on page 128</td>
</tr>
<tr>
<td>Specify the name of the subgroup, a collection of Q subscriptions between servers that are used for multidirectional replication</td>
<td>&quot;SET SUBGROUP command (multidirectional Q replication)&quot; on page 129</td>
</tr>
<tr>
<td>Specify the tables that participate in a bidirectional or peer-to-peer configuration</td>
<td>&quot;SET TABLES command (multidirectional Q replication)&quot; on page 130</td>
</tr>
<tr>
<td>Enable and disable the trace for the ASNCLP commands</td>
<td>SET TRACE command</td>
</tr>
<tr>
<td>Display the environment set during the session</td>
<td>SHOW SET ENV command</td>
</tr>
<tr>
<td>Start a Q subscription</td>
<td>START QSUB command</td>
</tr>
<tr>
<td>Stop a Q subscription</td>
<td>STOP QSUB command</td>
</tr>
<tr>
<td>Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions.</td>
<td>VALIDATE WSMQ ENVIRONMENT FOR command</td>
</tr>
<tr>
<td>Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map.</td>
<td>VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command</td>
</tr>
</tbody>
</table>
Chapter 4. ASNCLP commands for event publishing

The ASNCLP commands for event publishing define and change publishing queue maps and publications. The commands also can be used to start and stop publications.

“Sample ASNCLP scripts for setting up event publishing” on page 136 demonstrates how you can combine event publishing commands to create an ASNCLP setup script.

Table 14 lists the ASNCLP commands for event publishing and links to topics that describe each command.

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a column to a publication</td>
<td>ALTER ADD COLUMN command</td>
</tr>
<tr>
<td>Change a publishing queue map</td>
<td>“ALTER PUBQMAP command” on page 139</td>
</tr>
<tr>
<td>Change a publication</td>
<td>“ALTER PUB command” on page 140</td>
</tr>
<tr>
<td>Create the control tables for the Q Capture program</td>
<td>CREATE CONTROL TABLES FOR command</td>
</tr>
<tr>
<td>Create a publishing queue map</td>
<td>“CREATE PUBQMAP command” on page 142</td>
</tr>
<tr>
<td>Create a publication</td>
<td>“CREATE PUB command” on page 144</td>
</tr>
<tr>
<td>Drop the control tables for the Q Capture program</td>
<td>DROP CONTROL TABLES ON command</td>
</tr>
<tr>
<td>Delete a publishing queue map</td>
<td>“DROP PUBQMAP command” on page 145</td>
</tr>
<tr>
<td>Delete a publication</td>
<td>“DROP PUB command” on page 148</td>
</tr>
<tr>
<td>List publications</td>
<td>“LIST PUBS command” on page 149</td>
</tr>
<tr>
<td>List publishing queue maps</td>
<td>“LIST PUBQMAPS command” on page 150</td>
</tr>
<tr>
<td>List Q Capture schemas</td>
<td>LIST CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Promote a publication</td>
<td>“PROMOTE PUB command” on page 151</td>
</tr>
<tr>
<td>Promote a publishing queue map</td>
<td>“PROMOTE PUBQMAP command” on page 152</td>
</tr>
<tr>
<td>Start a publication</td>
<td>“START PUB command” on page 153</td>
</tr>
<tr>
<td>Set the Q Capture schema for all task commands</td>
<td>SET CAPTURE SCHEMA command</td>
</tr>
<tr>
<td>Define the log file for the ASNCLP program</td>
<td>SET LOG command</td>
</tr>
<tr>
<td>Define output files that contain SQL statements to set up event publishing</td>
<td>SET OUTPUT command</td>
</tr>
<tr>
<td>Set the WebSphere® MQ queue manager</td>
<td>SET QMANAGER command</td>
</tr>
<tr>
<td>Specify whether to automatically run each task command from an input file before the ASNCLP program processes the next task command</td>
<td>SET RUN SCRIPT command</td>
</tr>
<tr>
<td>Specify the Q Capture server to use in the ASNCLP session</td>
<td>SET SERVER command</td>
</tr>
<tr>
<td>Enable and disable the trace for the ASNCLP commands</td>
<td>SET TRACE command</td>
</tr>
<tr>
<td>Display the environment set during the session</td>
<td>SHOW SET ENV command</td>
</tr>
<tr>
<td>Stop a publication</td>
<td>“STOP PUB command” on page 153</td>
</tr>
</tbody>
</table>
Table 14. ASNCLP commands for event publishing (continued)

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verify that the required WebSphere MQ objects exist</td>
<td><strong>VALIDATE WSMQ ENVIRONMENT FOR command</strong></td>
</tr>
<tr>
<td>and have the correct properties for schemas, queue</td>
<td></td>
</tr>
<tr>
<td>maps, and publications.</td>
<td></td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up event publishing

This sample contains four ASNCLP scripts for setting up a basic event publishing environment. It includes Q Capture control tables, a publishing queue map, and an publication.

ASNCLP scripts typically generate one or more SQL scripts to create publishing objects. Because some publishing objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Q Capture control tables
2. Publishing queue map
3. Q subscription

The final ASNCLP script checks the WebSphere MQ environment for event publishing and does not generate SQL statements. You must create the control tables and queue map before you run this script.

This sample has a section for each ASNCLP script, which you can copy to a text file and run by using the `ASNCLP -f filename` command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

*Table 15 on page 138* below the sample describes the SQL scripts that are generated.

For help creating the WebSphere MQ objects that are used in these scripts, see [Graphical tool for generating WebSphere MQ setup scripts for Q replication and event publishing](#) and [WebSphere MQ setup scripts for Q replication](#).

**ASNCLP script 1 (Q Capture control tables)**

This script generates SQL statements that create Q Capture control tables at the SAMPLE database. It includes commands for the following tasks:

1. Setting the environment
2. Creating Q Capture control tables
3. Ending the ASNCLP session

   # Setting the environment.
   # The SET LOG command directs ASNCLP messages to one log file, ep.err.
   # The SET OUTPUT command creates an SQL script, epcontrol.sql.
   # The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
   # they are run.

   **ASNCLP SESSION SET TO Q REPLICATION;**
   SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
   SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
   SET CAPTURE SCHEMA SOURCE EPI;
   SET OUTPUT CAPTURE SCRIPT "epcontrol.sql";
SET RUN SCRIPT LATER;

# 2  Creating Q Capture control tables.
# The command specifies a restart queue and Q Capture administration queue.
# It also reduces the time that the Q Capture program pauses after reaching the
# end of the DB2 recovery log from the default of 5000 milliseconds (5 seconds)
# to 1000 milliseconds.

CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "EP1.QM1.RESTARTQ" ADMINQ "EP1.QM1.ADMINQ"
SLEEP INTERVAL 1000;

# 3  Ending the ASNCLP session.
QUIT;

**ASNCLP script 2 (publishing queue map)**

This script generates SQL statements that create a publishing queue map. It includes commands for the following tasks:

1. Setting the environment
2. Creating a publishing queue map
3. Ending the ASNCLP session

# 1  Setting the environment.
# The SET OUTPUT command creates a SQL script, pqmap.sql, which adds definitions
# for the queue map to the Q Capture control tables.

ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "pqmap.err";
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE EP1;
SET OUTPUT CAPTURE SCRIPT "pqmap.sql";
SET RUN SCRIPT LATER;

# 2  Creating a publishing queue map.
# This command generates SQL to create a publishing queue map,
# SAMPLE ASN1 TO SUBSCRIBER. It specifies a send queue at the Q Capture
# server. The command also specifies that the content of each message will
# be a single row, that the memory buffer for each message (MAX MESSAGE SIZE)
# will be 128 KB (double the default), and that that heartbeat messages will be
# sent every 5 seconds.
CREATE PUBQMAP SAMPLE ASN1 TO SUBSCRIBER USING
SENDQ "EP1.QM1.PUBDATAQ" MESSAGE CONTENT TYPE R
MAX MESSAGE SIZE 128 HEARTBEAT INTERVAL 5;

# 5  Ending the ASNCLP session.
QUIT;

**ASNCLP script 3 (XML publication)**

This script generates SQL statements to create an XML publication. It specifies a source table, DEPARTMENT, at the SAMPLE database. The script includes commands for the following tasks:

1. Setting the environment
2. Creating an XML publication
3. Ending the ASNCLP session

# 1  Setting the environment.
# The SET OUTPUT command creates a SQL script, pub.sql, that adds definitions
# for the XML publication to the Q Capture control tables.
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "pub.err";
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE EPI;
SET OUTPUT CAPTURE SCRIPT "pub.sql";
SET RUN SCRIPT LATER;

# 2 Creating the publication
# This command generates SQL to create an XML publication named DEPARTMENT0001.
# It specifies the DEPARTMENT table as a source. Messages will be sent when any
# column in the source table changes. DELETE operations at the source table will
# not prompt a message to be sent.
CREATE XML PUB USING PUBQMAP SAMPLE_EP1_TO_SUBSCRIBER
(PUBNAME "DEPARTMENT0001" DB2ADMIN.DEPARTMENT ALL CHANGED ROWS Y
SUPPRESS DELETES Y);

# 5 Ending the ASNCLP session.
QUIT;

ASNCLP script 4 (check WebSphere MQ environment)

This script does not generate SQL. Instead, it checks whether the queue manager
and queues that were specified in the other scripts exist and whether the objects
have the correct properties for event publishing. The script includes commands for
the following tasks:

1 Setting the environment
2 Checking the queue managers and queues
3 Ending the ASNCLP session

# 1 Setting the environment.
# No SET RUN statement is required. The commands run immediately and send results
# to the command window and log.
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "epchecks.err";
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
SET OUTPUT CAPTURE SCRIPT EPI;

# 2 Checking the queue manager and queues.
# These commands check whether the queue manager and queues exist and validate
# their settings. Any errors must be corrected before you start the Q Capture
# program.
VALIDATE WSMQ ENVIRONMENT FOR CAPTURE SCHEMA;
VALIDATE WSMQ ENVIRONMENT FOR PUBQMAP SAMPLE_ASN1_TO_SUBSCRIBER;

# 3 Ending the ASNCLP session.
QUIT;

Output of the script

In addition to the log file, ep.err, this example produces three SQL script files in
the same directory where you run the ASNCLP program. Table 15 describes the
files.

Table 15. SQL script files that are created by the sample ASNCLP scripts

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>epccontrol.sql</td>
<td>Create Q Capture control tables</td>
</tr>
</tbody>
</table>
Table 15. SQL script files that are created by the sample ASNCLP scripts (continued)

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to...</th>
</tr>
</thead>
<tbody>
<tr>
<td>pqmap.sql</td>
<td>Create a publishing queue map</td>
</tr>
<tr>
<td>pub.sql</td>
<td>Create a publication</td>
</tr>
</tbody>
</table>

**ALTER PUBQMAP command**

Use the ALTER PUBQMAP command to change attributes for an existing publishing queue map.

**Syntax**

```
ALTER PUBQMAP qmapname USING options
```

**options:**

- **DESC** "description"  
- **MESSAGE CONTENT TYPE** T | R  
- **SENDQ** sendqname

- **ERROR ACTION** I | S  
- **HEARTBEAT INTERVAL** interval  
- **MAX MESSAGE SIZE** size

- **HEADER** NONE  
- ON CODEPAGE ERROR SEND NO DATA | SEND RAW DATA

**Parameters**

- **qmapname**  
  Specifies the name of the publishing queue map.

- **DESC "description"**  
  Specifies the description of the publishing queue map.

**MESSAGE CONTENT TYPE**

Specifies whether messages put on the queue will contain an entire database transaction or only a row operation.

- **T**  
  Messages contain all of the row operations (update, insert, or delete) within a DB2 transaction, and information about the transaction. This is the default.

- **R**  
  Messages contain a single update, insert, or delete operation, and information about the DB2 transaction to which it belongs.

- **SENDQ sendqname**  
  Specify to updates the send queue used by the publishing queue map.

**ERROR ACTION**

Specify to tell the Q Capture program what to do when the send queue stops accepting messages. For example, the queue might be full, or the queue manager might have reported a severe error for this queue.
The Q Capture program invalidates all Q subscriptions and publications for this queue but continues to put messages on other queues. This is the default.

The Q Capture program stops when an error is detected on this queue.

**HEARTBEAT INTERVAL**  \(\text{interval}\)
- Specifies the interval (in seconds) between heartbeat messages sent by the Q Capture program to a subscribing application when there are no transactions to publish.

**MAX MESSAGE SIZE**  \(\text{size}\)
- Specifies the maximum size (in kilobytes) of the buffer that is used for sending messages over the send queue.

**HEADER**
- Specifies whether you want a JMS-compliant MQ RFH2 header added to all messages that use the send queue that is specified in this publishing queue map.
  - **NONE**
    - Specify to send only the publication message with no special headers.
  - **MQ RFH2**
    - Specify to attach a special header to the message that will contain the topic name that you specify as part of an publication.

**ON CODEPAGE ERROR**
- Specifies whether you want to send data when code page conversion errors occur.
  - **SEND NO DATA**
    - The Q Capture program does not send the data when an error occurs during code page conversion.
  - **SEND RAW DATA**
    - The Q Capture program sends hex representation of the character data if a code page conversion error occurs.

**Example**

To alter the SAMPLE ASN1_TO_SUBSCRIBER publishing queue map and change the message type from row to transaction, stop the Q Capture program if an error occurs, specify 6 seconds between heartbeat messages, and set the maximum size of the buffer to 64 kilobytes for sending messages over the send queue:

```
ALTER PUBQMAP SAMPLE ASN1 TO SUBSCRIBER USING MESSAGE CONTENT TYPE T ERROR ACTION S
HEARTBEAT INTERVAL 6 MAX MESSAGE SIZE 64
```

**ALTER PUB command**

Use the ALTER PUB command to alter an publication.

**Syntax**

```
ALTER PUB  pubname FOR source_name [source_owner,] DESC "description"
```
Parameters

PUB pubname
   Specifies the name of the publication.

source_owner
   Specifies the source table schema.

source_name
   Specifies the source table name.

DESC "description"
   Specifies a description of the publication.

PUBQMAP qmapname
   Specifies the new name of the publishing queue map that is used by this publication.

other-opt-clause:

SEARCH CONDITION "search_cond"
   Specifies a search condition for filtering changes to publish. The change is not sent if the predicate is false. This is an annotated select WHERE clause, where there must be a colon before the column names of the source table. The following example shows a WHERE clause:
   
   ALTER PUB mypubname FOR ALLTYPE1 OPTIONS SEARCH CONDITION "WHERE :MYKEY > 1000"

ALL CHANGED ROWS
   Specifies a data sending option.
     Y  Send a row when any column in the source table changes.
     N  Send a row only if a subscribed column in the source table changes.

BEFORE VALUES
   For an update operation, this keyword indicates whether the Q Capture program sends the before values of non-key columns in addition to their after values. For a delete, this keyword indicates whether the Q Capture program sends the before values of non-key columns in addition to the before values of the key columns.
     N  The Q Capture program does not send before values of non-key columns that change. If a key column changes, the Q Capture program sends both its before and after values. For delete statements involving key columns, only before values are sent. This is the default.
Y When there are changes to non-key columns in the source table that are part of a publication, the Q Capture program sends both before and after values.

**CHANGED COLS ONLY**
Specifies whether the Q Capture program publishes columns that are part of a publication only if they have changed. This keyword only applies to update operations.

Y When the Q Capture program sends an updated row, it sends only the changed columns that are part of a publication. This is the default.

N The Q Capture program sends all columns in a row that are part of a publication whenever any of them have changed.

**SUPPRESS DELETES**
Specifies whether to send rows that were deleted from the source table.

N Send deleted rows.

Y Do not send deleted rows.

**TOPIC "topic"**
Specifies the topic that will be included in the MQ RFH2 message header and used by the publication. You must specify the HEADER MQ RFH2 keywords when you create the publishing queue map that this publication uses.

**Example**
To alter the publication MYXMLPUB by only sending a row if the subscribed column has changed, sending all columns in a row that are part of the publication whenever any of them have changed, and sending deleted rows:

```
ALTER PUB MYXMLPUB FOR ERIC.TSTTABLE OPTIONS ALL CHANGED ROWS N BEFORE VALUES N CHANGED COLS ONLY N SUPPRESS DELETES N
```

**CREATE PUBQMAP command**
Use the CREATE PUBQMAP command to create a publishing queue map that specifies the send queue to use for event publishing and whether to send messages in XML or delimited format.

**Syntax**

```
CREATE PUBQMAP qmapname DESC "description" USING SENDQ "sendqname"
```

```
MESSAGE FORMAT XML xml-format-options
```

```
DELIMITED del-format-options
```

**xml-format-options:**

```
MESSAGE CONTENT TYPE T R ERROR ACTION I S HEARTBEAT INTERVAL interval
```
Parameters

qmapname
   Specifies the name of the publishing queue map.

DESC "description"
   Specifies the description of the publishing queue map.

SENDQ "sendqname"
   Specifies the name of the WebSphere MQ queue to use as the send queue.

MESSAGE FORMAT
   Specifies whether you want to publish messages in XML format or delimited format. Use this keyword if you want to specify options for the publishing queue map; the options differ for the different message format types.

MESSAGE CONTENT TYPE
   Specifies whether messages put on the queue will contain an entire database transaction or only a row operation.
   T    Messages contain all of the row operations (update, insert, or delete) within a DB2 transaction, and information about the transaction. This is the default.
   R    Messages contain a single update, insert, or delete operation, and information about the DB2 transaction to which it belongs.

ERROR ACTION
   Specify to tell the Q Capture program what to do when the send queue stops accepting messages. For example, the queue might be full, or the queue manager might have reported a severe error for this queue.
   I    This value is deprecated. If you specify I, the Q Capture program will act as though S were specified and stop.
   S    The Q Capture program stops.

HEARTBEAT INTERVAL interval
   XML format only: Specifies the interval (in seconds) between heartbeat messages that are sent by the Q Capture program to a subscribing application when there are no transactions to publish. To disable heartbeat messages, set the heartbeat interval to 0. Heartbeat messages are not supported for the delimited message format, so the value of this keyword is always 0 for delimited format.

MAX MESSAGE SIZE size
   Specifies the maximum size (in kilobytes) of the buffer used for sending messages over the send queue.
HEADER
Specifies whether you want a JMS-compliant MQ RFH2 header added to all messages that use the send queue that is specified in this publishing queue map. This keyword is not supported for delimited message format.

NONE
Specify to send only the publication message with no special headers.

MQ RFH2
Specify to attach a special header to the message that will contain the topic name that you specify as part of an publication.

ON CODEPAGE ERROR
Specifies whether you want to send data when code page conversion errors occur.

SEND NO DATA
The Q Capture program does not send the data when an error occurs during code page conversion.

SEND RAW DATA
The Q Capture program sends hex representation of the character data if a code page conversion error occurs.

Example 1
To create a publishing queue map SAMPLE_ASN1_TO_SUBSCRIBER that sets the message content type to row, specifies 5 seconds between heartbeat messages, and sets a maximum message size of 128 KB:

CREATE PUBQMAP SAMPLE_ASN1_TO_SUBSCRIBER USING
SENDQ "ASN1.QM1.PUBDATAQ" MESSAGE CONTENT TYPE R
HEARTBEAT INTERVAL 5 MAX MESSAGE SIZE 128

Example 2
To create a publishing queue map SAMPLE_ASN_TO_DATASTAGE that sets the message format to delimited, the message type to row, and a maximum message size of 256 KB:

CREATE PUBQMAP SAMPLE_ASN_TO_DATASTAGE
USING SENDQ "ASN.QM1.DELIMDATAQ" MESSAGE FORMAT DELIMITED
MESSAGE CONTENT TYPE R HEARTBEAT INTERVAL 0 MAX MESSAGE SIZE 256

CREATE PUB command
Use the CREATE PUB command to create a publication.

Syntax

```
CREATE PUB [USING PUBQMAP qmapname]
```

```
(src-clause)
```

```
(PUBNAME pubname)
```

```
(DESC description)
```

```
(PUBQMAP qmapname)
```

```
(opt-clause)
```

```
)
```
src-clause:

```
[ source_owner, ]
SRC OWNER LIKE " predicate1 "
SRC NAME LIKE " predicate2 "
SRC ALL

COLS col-cause
```

col-cause:

```
ALL

INCLUDE ( colname )
EXCLUDE ( colname )

ISKEY ( colname )
```

opt-clause:

```
SEARCH CONDITION " search_cond "
ALL CHANGED ROWS N Y

BEFORE VALUES N Y

INCLUDE ( colname )

CHANGED COLS ONLY Y N

HAS LOAD PHASE N Y

SUPPRESS DELETES N Y

TOPIC " topic "
```

Parameters

**USING PUBQMAP qmapname**
- Specifies the publishing queue map that is used by all subsequent publications that are created by this command.

**PUBNAME pubname**
- Specifies the name of the publication.

**DESC "description"**
- Specifies a description of the publication.

**PUBQMAP qmapname**
- Specifies the publishing queue map that is used by this publication. If you do not specify the **USING PUBQMAP** keyword, you must define the **PUBQMAP** keyword for every publication that you define.
source_owner
 specifies the schema of the source table.

source_name
 specifies the name of the source table.

SRC OWNER LIKE "predicate1"
 specify to choose all tables with a schema that matches the expression in the LIKE statement. The following examples show LIKE statements:
CREATE PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "ASN%");
CREATE PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%);

SRC NAME LIKE "predicate2"
 specify to choose all tables with a name that matches the expression in the LIKE statement. The following example shows a LIKE statement:
CREATE PUB USING PUBQMAP ABCDPUBQMAP
(SRC NAME LIKE "%4%")

SRC ALL
 specify to choose all tables, with the exception of DB2 catalog views, that exist on the Q Capture server.

col-cause:

ALL
 specify to publish all columns in the source table.

INCLUDE (colname)
 specifies what columns to publish. You can specify multiple columns.

EXCLUDE (colname)
 specifies what columns not to publish. You can specify multiple columns.

ISKEY (colname)
 indicates whether the column is part of the key to use for publishing. Any column or set of columns that are unique at the source can be used. If no column is specified as a key, the Q Capture program looks for a primary key within the set of published columns, then for a unique constraint, and then for a unique index. If none of these exists, Q Capture will use all published, valid columns as key columns for publishing. (Some columns, such as LOB columns, cannot be used as keys.)

opt-clause:

SEARCH CONDITION "search_cond"
 specifies a search condition for filtering changes to publish. The change is not sent if the predicate is false. This is an annotated select WHERE clause, which requires a colon before the column names. The following example shows a WHERE clause:
CREATE PUB USING PUBQMAP ASNMAP
(PUBNAME mypubname ALLTYPE1 SEARCH CONDITION
"WHERE :MYKEY > 1000")

ALL CHANGED ROWS
 specifies a data sending option.

Y
 send a row when any column in the source table changes.

N
 send a row only if a subscribed column in the source table changes.
BEFORE VALUES
For an update operation, this keyword indicates whether the Q Capture program sends the before values of non-key columns in addition to their after values. For a delete, this keyword indicates whether the Q Capture program sends the before values of non-key columns in addition to the before values of the key columns.

N The Q Capture program does not send before values of nonkey columns that change. If a key column changes, the Q Capture program sends both its before and after values. For delete statements involving key columns, only before values are sent. This is the default.

Y When there are changes to nonkey columns in the source table that are part of a publication, the Q Capture program sends both before and after values.

INCLUDE (colname)
Specifies the nonkey columns for which the Q Capture program sends both before and after values.

CHANGED COLS ONLY
This keyword indicates whether the Q Capture program publishes columns that are part of a publication only if they have changed. This field applies to update operations only.

Y When the Q Capture program sends an updated row, it sends only the changed columns that are part of a publication. This is the default.

N The Q Capture program sends all columns in a row that are part of a publication whenever any of them has changed.

HAS LOAD PHASE
Specifies whether the target table for the publication will be loaded with data from the source.

N No load phase at the target. This is the default.

E External load: Specifies a manual load by an application outside of replication. In this case, you insert the LOADDONE signal (using the LOADDONE command) into the IBMQREP_SIGNAL table at the Q Capture server to inform the Q Capture program that the application is done loading.

SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table.

N Send deleted rows.

Y Do not send deleted rows.

TOPIC "topic"
Specifies the topic that will be included in the MQ RFH2 message header and used by the publication. You must specify the HEADER MQ RFH2 keywords when you create the publishing queue map that this publication uses.

Example 1
To create a publication using publishing queue map
SAMPLE ASN1 TO SUBSCRIBER that publishes a row when any column in the source table changes and does not publish rows that were deleted from the source table:
CREATE PUB USING PUBQMAP SAMPLE_ASN1_TO_SUBSCRIBER (PUBNAME "EMPLOYEE0001"
DB2ADMIN.EMPLOYEE ALL CHANGED ROWS Y BEFORE VALUES Y CHANGED COLS ONLY Y
HAS LOAD PHASE N SUPPRESS DELETES Y)

Example 2

To create a publication and specify that the capture program sends before values
for the nonkey columns C10, C11, and C13:
ASNCLP SESSION SET TO Q REPLICATION;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
SET SERVER CAPTURE TO DB APP1DB;
SET CAPTURE SCHEMA SOURCE SAMPLE;
CREATE PUB USING PUBQMAP 'PUBQ1' (PUBNAME PUB1 DATA.EMPLOYEE
OPTIONS BEFORE VALUES INCLUDE(c10, c11, c12));

DROP PUBQMAP command

Use the DROP PUBQMAP command to delete an existing publishing queue map.

Restriction: The publications that are using the publishing queue map must first
be deleted.

Syntax

```
DROP PUBQMAP qmapname
```

Parameters

qmapname

Specifies the name of the publishing queue map to drop.

Example

To drop the SAMPLE_ASN1_TO_SUBSCRIBER publishing queue map:
DROP PUBQMAP SAMPLE_ASN1_TO_SUBSCRIBER

DROP PUB command

Use the DROP XML PUB command to delete a publication.

Syntax

```
DROP PUB
```

Parameters

ALL

Specify to delete all of the publications for the schema and server set through
the SET commands.
**PUBNAME** *pubname*
Specify the name of an publication to delete.

**FOR PUBNAME LIKE** "*predicate*"
Specify to delete all publications that match the LIKE statement. The following example shows a LIKE statement:
```
DROP PUB FOR PUBNAME LIKE "pubname02%"
```

**Example**
To delete a publication:
```
DROP PUB (PUBNAME MYPUB)
```

---

**LIST PUBS command**

You can use the LIST PUBS command to list publications for a specified Q Capture server or schema.

**Syntax**
```
LIST PUBS [FOR SCHEMA *schema*] [SERVER *dbparms*]
```

**dbparms-clause:**
```
[DBALIAS *aliasname*] [DBNAME *dbname*] [ID *userid*] [PASSWORD *pwd*] [FILE *filename*] [CONFIG SERVER *servername*]
```

**Parameters**

**FOR SCHEMA** *schema*
- Specifies which schema to use. The default is "ASN".

**dbparms-clause:**

**SERVER**
- Specifies the server containing the publications to list.

**DB** *dbalias*
- Specifies the database alias name.

**DBALIAS** *aliasname*
- Specifies the database alias name.

**DB NAME** *dbname*
- Specifies the database name.

**ID** *userid*
- Specifies the user ID to use to connect to the database.

**PASSWORD** *pwd*
- Specifies the password to use for connections.

**CONFIG SERVER** *servername*
**Classic sources:** Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP should use to connect to the Classic server.

**FILE filename**  
Specifies the complete path and file name to the replication configuration file.  
If you do not use the FILE parameter, the ASNCLP attempts to use the asn_servers.ini file in the current directory, if that file exists. Use the FILE parameter with different files that are customized for different environments.

---

**LIST PUBQMAPS command**

You can use the LIST PUBQMAPS command to list publication queue maps for a specified Q Capture server or schema.

**Syntax**

```plaintext
LIST PUBQMAPS FOR SCHEMA schema [SERVER dbparms]
```

**dbparms-clause:**

```plaintext

DBALIAS aliasname
DBNAME dbname
CONFIG SERVER servername
FILE filename
ID userid
PASSWORD pwd
```

**Parameters**

**FOR SCHEMA schema**  
Specifies which schema to use. The default is "ASN".

**dbparms-clause:**

**SERVER**  
Specifies the server containing the publishing queue maps to list.

**DB dbalias**  
Specifies the database alias name.

**DBALIAS aliasname**  
Specifies the database alias name.

**z/OS DBNAME dbname**  
Specifies the database name.

**ID userid**  
Specifies the user ID to use to connect to the database.

**PASSWORD pwd**  
Specifies the password to use for connections.

**CONFIG SERVER servername**

**Classic sources:** Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP should use to connect to the Classic server.

**FILE filename**  
Specifies the complete path and file name to the replication configuration file.
If you do not use the FILE parameter, the ASNCLP attempts to use the
asnserver.ini file in the current directory, if that file exists. Use the FILE
parameter with different files that are customized for different environments.

PROMOTE PUB command

Use the PROMOTE PUB command to promote the definitions of one or more
publications. You can use this command to customize the properties of the
publication such as the name of the publication and the name of the send queue.
The values for other properties are set to the same values as the current
publication.

You can use the ALTER PUB command to change other properties after you
promote the publication.

Syntax

```
PROMOTE PUB NAME (pubname) LIKE "predicate" FOR PUBQMAP pubqmap
```

Parameters

NAME pubname
   Specifies one or more publication names to promote. Separate multiple
   publication names with a comma.

LIKE "predicate"
   Specifies part of a publication name to promote. All publications matching this
   predicate are promoted.

FOR PUBQMAP pubqmap
   Specifies an existing publishing queue map. All publications using that
   publishing queue map are promoted.

Example - matching a predicate

To promote all publications that start with the name EMP:
```
PROMOTE PUB NAME (EMPLOYEE021,EMPLOYEE032);
```

Example - using a publishing queue map

To promote all publications that use the qmap1 publishing queue map:
```
PROMOTE PUB NAME (EMPLOYEE021,EMPLOYEE032);
```

Example - naming publications

To promote publications that are named EMPLOYEE021 and EMPLOYEE032:
```
PROMOTE PUB NAME (EMPLOYEE021,EMPLOYEE032);
```
PROMOTE PUBQMAP command

Use the PROMOTE PUBQMAP command to promote the definitions of one or more publishing queue maps from one set of control tables to another set of control tables. You can also use this command to change some properties when the publishing queue map is promoted, such as the name of the send queue and name of the publishing queue map. The promoted values of properties that cannot be customized are taken from the source publishing queue map. If you need to change other properties, you can use the ALTER PUBQMAP command after promoting the publishing queue map to change the properties for the new publishing queue map.

Syntax

```
>>> PROMOTE PUBQMAP NAME pubqmapname LIKE "predicate" USING new-clause
```

new-clause:

```
PUBQMAP new-qmap
SENDQ new-sendq
```

Parameters

**NAME pubqmapname**
- Specifies the name of an existing publishing queue map to be promoted.

**USING**
- Specifies new values for properties for the promoted publishing queue map.

**LIKE "predicate"**
- Promotes all publishing queue maps that match the predicate name. You cannot customize the properties if you use this option.

new-clause:

**PUBQMAP new-qmap**
- Specifies the name of the publishing queue map. If you do not specify a name, the current publishing queue map name is used.

**SENDQ new-sendq**
- Specifies the send queue of the promoted publishing queue map. If you do not specify a send queue name, the current send queue name is used.

Usage notes

- You must use the SET SERVER command with the PROMOTE option to set the environment for your promotions. The SET SERVER command allows you to specify the server that contains the publishing queue map to be promoted and to define which server the publishing queue map is promoted to.

- You cannot change the values for some properties by using the PROMOTE PUBQMAP command. You can later use the ALTER PUBQMAP command to change the value for other properties after you promote the publishing queue map.
Example 1

To promote all publishing queue maps that match the name "SAMPLE_ASN%":

```
PROMOTE PUBQMAP LIKE "SAMPLE_ASN%";
```

Example 2

To promote publishing queue map PUBQMAP2, and change the name of the publishing queue map to pubqmapnew and change the name of the send queue to sendqnew2:

```
PROMOTE PUBQMAP NAME PUBQMAP2 USING PUBQMAP pubqmapnew SENDQ "sendqnew2";
```

### START PUB command

Use the START PUB command to start a publication.

**Syntax**

```
START PUB PUBNAME pubname
  FOR PUBNAME LIKE "predicate"
```

**Parameters**

- **PUBNAME pubname**
  Specifies the name of the publication to start.

- **FOR PUBNAME LIKE "predicate"**
  Specify to start publications that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
START PUB FOR PUBNAME LIKE "%table%"
```

**Example**

To start a publication:

```
START PUB PUBNAME MYXMLPUB
```

### STOP PUB command

Use the STOP PUB command to stop a publication.

**Syntax**

```
STOP PUB PUBNAME pubname
  FOR PUBNAME LIKE "predicate"
```

**Parameters**

- **PUBNAME pubname**
  Specifies the name of the publication to stop.

- **FOR PUBNAME LIKE "predicate"**
  Specify to stop publications that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
STOP PUB FOR PUBNAME LIKE "%table%"
```
Example

To stop a publication:
STOP PUB PUBNAME MYPUB
Chapter 5. ASNCLP commands for the Replication Alert Monitor

The ASNCLP commands for the Replication Alert Monitor define and change objects such as control tables, contacts, alert conditions, and suspensions.

“Sample ASNCLP scripts for setting up the Replication Alert Monitor” on page 156 demonstrates how you can combine Replication Alert Monitor commands to create an ASNCLP setup script.

Table 16 lists the ASNCLP commands for the Replication Alert Monitor and links to topics that describe each command.

Table 16. ASNCLP commands for the Replication Alert Monitor

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change alert conditions for the Apply program</td>
<td>ALTER ALERT CONDITIONS FOR APPLY command” on page 159</td>
</tr>
<tr>
<td>Change alert conditions for the Capture program</td>
<td>ALTER ALERT CONDITIONS FOR CAPTURE command’ on page 161</td>
</tr>
<tr>
<td>Change alert conditions for the Q Apply program</td>
<td>ALTER ALERT CONDITIONS FOR QAPPLY command” on page 164</td>
</tr>
<tr>
<td>Change alert conditions for the Q Capture program</td>
<td>ALTER ALERT CONDITIONS FOR QCAPTURE command” on page 166</td>
</tr>
<tr>
<td>Change contact information for notifications</td>
<td>ALTER CONTACT command” on page 168</td>
</tr>
<tr>
<td>Change a contact group</td>
<td>ALTER GROUP command” on page 169</td>
</tr>
<tr>
<td>Change a monitor suspension</td>
<td>ALTER MONITOR SUSPENSION command” on page 169</td>
</tr>
<tr>
<td>Change a monitor suspension template</td>
<td>ALTER MONITOR SUSPENSION TEMPLATE command” on page 170</td>
</tr>
<tr>
<td>Create alert conditions for the Apply program</td>
<td>CREATE ALERT CONDITIONS FOR APPLY command” on page 171</td>
</tr>
<tr>
<td>Create alert conditions for the Capture program</td>
<td>CREATE ALERT CONDITIONS FOR CAPTURE command’ on page 174</td>
</tr>
<tr>
<td>Create alert conditions for the Q Apply program</td>
<td>CREATE ALERT CONDITIONS FOR QAPPLY command” on page 175</td>
</tr>
<tr>
<td>Create alert conditions for the Q Capture program</td>
<td>CREATE ALERT CONDITIONS FOR QCAPTURE command” on page 177</td>
</tr>
<tr>
<td>Create contact information for notifications</td>
<td>CREATE CONTACT command” on page 179</td>
</tr>
<tr>
<td>Create the control tables for the Monitor program</td>
<td>CREATE CONTROL TABLES FOR command” on page 180</td>
</tr>
<tr>
<td>Create a contact group</td>
<td>CREATE GROUP command” on page 182</td>
</tr>
<tr>
<td>Create a monitor suspension</td>
<td>CREATE MONITOR SUSPENSION command” on page 182</td>
</tr>
<tr>
<td>Create a monitor suspension template</td>
<td>CREATE MONITOR SUSPENSION TEMPLATE command” on page 184</td>
</tr>
<tr>
<td>Delegate an existing contact to a new contact</td>
<td>DELEGATE CONTACT command” on page 185</td>
</tr>
</tbody>
</table>
Table 16. ASNCLP commands for the Replication Alert Monitor (continued)

<table>
<thead>
<tr>
<th>If you want to ...</th>
<th>Use this command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete alert conditions for the Apply program</td>
<td>“DROP ALERT CONDITIONS FOR APPLY command” on page 185</td>
</tr>
<tr>
<td>Delete alert conditions for the Capture program</td>
<td>“DROP ALERT CONDITIONS FOR CAPTURE command” on page 186</td>
</tr>
<tr>
<td>Delete alert conditions for the Q Apply program</td>
<td>“DROP ALERT CONDITIONS FOR QAPPLY command” on page 186</td>
</tr>
<tr>
<td>Delete alert conditions for the Q Capture program</td>
<td>“DROP ALERT CONDITIONS FOR QCAPTURE command” on page 186</td>
</tr>
<tr>
<td>Delete an existing contact</td>
<td>“DROP CONTACT command” on page 187</td>
</tr>
<tr>
<td>Delete a contact group</td>
<td>“DROP GROUP command” on page 187</td>
</tr>
<tr>
<td>Delete a monitor suspension</td>
<td>“DROP MONITOR SUSPENSION command” on page 188</td>
</tr>
<tr>
<td>Delete a monitor suspension template</td>
<td>“DROP MONITOR SUSPENSION TEMPLATE command” on page 188</td>
</tr>
<tr>
<td>List monitor suspensions</td>
<td>“LIST MONITOR SUSPENSION command” on page 189</td>
</tr>
<tr>
<td>List monitor suspension templates</td>
<td>“LIST MONITOR SUSPENSION TEMPLATE command” on page 189</td>
</tr>
<tr>
<td>Specify the server (database) used in the ASNCLP session, authentication information, and other required parameters for connecting to the server</td>
<td>“SET SERVER command” on page 190</td>
</tr>
<tr>
<td>Substitute one existing contact with another existing contact</td>
<td>“SUBSTITUTE CONTACT command” on page 191</td>
</tr>
</tbody>
</table>

Sample ASNCLP scripts for setting up the Replication Alert Monitor

This sample contains two ASNCLP scripts for setting up the Replication Alert Monitor. It includes Monitor control tables, a contact, and alert conditions.

In Classic replication, you can only monitor the Q Apply server.

ASNCLP scripts typically generate one or more SQL scripts to create replication objects. Because some replication objects depend on the existence of other objects, run the ASNCLP scripts and the SQL scripts that they generate in the following order:
1. Monitor control tables
2. Contact and alert conditions
3. Suspension template and suspension

Table 17 on page 159 below the sample describes each SQL script.

This sample has a section for each ASNCLP script, which you can copy to a text file and run by using the ASNCLP -f filename command. Within the code sample in each section, details about each group of commands are preceded by a comment character (#).

**ASNCLP script 1 (Monitor control tables)**

This script generates SQL statements that create Monitor control tables at the SAMPLE database. It includes commands for the following tasks:
Setting the environment
Creating Monitor control tables
Ending the ASNCLP session

# Setting the environment.
# The ASNCLP SESSION SET command is not required for the monitor because
# the session must be set to SQL replication and this is the default if no
# command is entered.
# The SET LOG command directs ASNCLP messages to one log file, moncontrol.err.
# The SET OUTPUT command creates an SQL script, moncontrol.sql.
# The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
# they are run.

SET OUTPUT MONITOR SCRIPT "moncontrol.sql";
SET LOG "moncontrol.err";
SET SERVER MONITOR TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT LATER;

# Creating Monitor control tables.

CREATE CONTROL TABLES FOR MONITOR CONTROL SERVER
IN UW OTHERS TSMON1;

# Ending the ASNCLP session.

QUIT;

ASNCLP script 2 (contact and alert conditions)

This script generates SQL statements that define alert conditions for the Monitor
and a contact to be alerted when a condition is met. It includes commands for the
following tasks:

1. Setting the environment
2. Specifying a contact
3. Defining alert conditions
4. Ending the ASNCLP session

# Setting the environment
# Three SET SERVER commands are required in this script: You set the Monitor
# server to specify which set of Monitor control tables will store information
# about the contact and alert conditions. You set the Capture and target servers
# to specify which servers will be monitored for the alert conditions that you
# will define.
# The SET OUTPUT command creates an SQL script, conalert.sql.

SET OUTPUT MONITOR SCRIPT "conalert.sql";
SET LOG "conalert.err";
SET SERVER MONITOR TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER CAPTURE TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT LATER;

# Specifying a contact
# The CREATE CONTACT command defines a contact name and specifies that alerts
# be sent to an email address.

CREATE CONTACT repladmin EMAIL "repladmin@us.ibm.com" DESCRIPTION
"Replication administrator";

# Creating alert conditions.
# These commands create alert conditions for the Q Capture program that runs
# at the monitored server SAMPLE and the Q Apply program that runs at the
# monitored server TARGET. The Q Capture conditions trigger an alert if Q Capture
# is down or if any errors or warnings occur. The LATENCY condition triggers
# an alert if the average Q Capture latency exceeds 2 seconds. The Q Apply
# conditions trigger an alert if Q Apply is down or if any errors or warnings
# occur. The EXCEPTIONS condition triggers an alert if a row is added to the
# IBMQREP_EXCEPTIONS table, signaling a SQL error or conflict.

CREATE ALERT CONDITIONS FOR QCAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
NOTIFY CONTACT repladmin (STATUS DOWN, ERRORS, WARNINGS, LATENCY 2);
CREATE ALERT CONDITIONS FOR QAPPLY SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
NOTIFY CONTACT repladmin (STATUS DOWN, ERRORS, WARNINGS, EXCEPTIONS);

# 5 Ending the ASNCLP session.
QUIT;

**ASNCLP script 3 (suspension template and suspension)**

This script generates SQL statements that create a monitor suspension template to
define a repeating pattern of monitor suspensions, and a monitor suspension to
put the template into effect. It includes commands for the following tasks:

1. Setting the environment
2. Creating a monitor suspension template
3. Creating a monitor suspension
4. Ending the ASNCLP session

# 1 Setting the environment
# Two SET SERVER commands are required in this script: You set the Monitor
# server to specify which set of Monitor control tables will store information
# about the template and suspension. You set the Capture server to specify
# the server where monitoring will be periodically suspended.
# The SET OUTPUT command creates an SQL script, suspend.sql.

SET OUTPUT MONITOR SCRIPT "suspend.sql";
SET LOG "suspend.err";
SET SERVER MONITOR TO DB SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET SERVER APPLY TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT LATER;

# 2 Creating a suspension template
# The CREATE MONITOR SUSPENSION TEMPLATE command creates a template that suspends
# the monitor program during the lunch hour every day.

CREATE MONITOR SUSPENSION TEMPLATE LUNCH START TIME 12:00:00
REPEATS DAILY FOR DURATION 1 HOUR;

# 3 Creating a suspension
# The CREATE MONITOR SUSPENSION command creates a suspension names S1
# that specifies that monitoring at the TARGET database will be suspended
# every day from 2006-12-10 to 2007-12-31. The command uses the template
# LUNCH to start the suspension at 12:00:00 for a period of hour.

CREATE MONITOR SUSPENSION NAME S1 FOR SERVER TARGET STARTING DATE 2006-12-10
USING TEMPLATE LUNCH ENDING DATE 2007-12-31;

# 5 Ending the ASNCLP session.
QUIT;

**Output of the script**

In addition to the log file, monitor.err, this example produces three SQL script files
in the same directory where you run the ASNCLP program. Table 17 on page 159
describes the files and where they run.
Table 17. SQL script files that are created by the sample ASNCLP scripts

<table>
<thead>
<tr>
<th>Output file</th>
<th>Contains SQL to ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>moncontrol.sql</td>
<td>Create Monitor control tables</td>
</tr>
<tr>
<td>conalert.sql</td>
<td>Define a contact and alert conditions</td>
</tr>
<tr>
<td>suspend.sql</td>
<td>Create a suspension template and suspension</td>
</tr>
</tbody>
</table>

**ALTER ALERT CONDITIONS FOR APPLY command**

Use the ALTER ALERT CONDITIONS FOR APPLY command to alter alert conditions for the Apply program.

**Syntax**

```
ALTER ALERT CONDITIONS FOR APPLY —QUALIFIER—QUAL—name—
  SET NAME—set-name—

MONITOR-QUALIFIER—mon—qual—
  notify-clause—
    add-or-remove-clause—
      change-clause—

notify-clause:
  NOTIFY—CONTACT—contact-name—
    GROUP—group-name—
    OPERATOR CONSOLE—

add-or-remove-clause:
  ADD—STATUS DOWN—
    ERRORS—
    WARNINGS—
    SUBSCRIPTIONS FAILING—
    SUBSCRIPTIONS DELAYED—time—
    SUBSCRIPTIONS INACTIVE—
    SUBSCRIPTIONS REFRESHED—
    TRANSACTIONS REJECTED—
    REWORKED ROWS—rows—
    LATENCY—end-end-latency—

change-clause:
  CHANGE—SUBSCRIPTIONS DELAYED—time—
    REWORKED ROWS—rows—
    LATENCY—end-end-latency—
```
Parameters

APPLY QUALIFIER qual-name
   Specifies the Apply qualifier.

SET NAME set-name
   Specifies the subscription set name. If you do not specify a subscription set
   name, all of the set names in the Apply qualifier will be assumed.

MONITOR QUALIFIER mon-qual
   Specifies the Monitor qualifier.

NOTIFY
   Specifies the contact or group of contacts to notify when the alert condition
   occurs.

   CONTACT contact-name
      Specifies the contact to notify.

   GROUP group-name
      Specifies the group to notify.

OPERATOR CONSOLE
   [z/OS] Specifies that alert notifications are sent to the z/OS
   console. This option is valid only if the monitor server is on a z/OS
   subsystem.

ADD
   Specify to add an alert condition.

REMOVE
   Specify to remove an alert condition.

CHANGE
   Specify to change an alert condition.

STATUS DOWN
   Specifies whether the Monitor program uses the asncmd status command to
   verify that the Apply program is running. The asncmd status command uses
   the DB2 Administration Server for non-OS/400 systems. If the Apply program
   is not running, an alert is sent.

ERRORS
   Specifies that the Monitor program checks if any error messages were logged
   in the IBMSNAP_APPLYTRACE table, specifically, any rows that have a value
   of ERROR for the OPERATION column. If any row is fetched, the
   DESCRIPTION column is included in the alert.

WARNINGS
   Specifies that the Monitor program checks if any warnings were logged in
   the IBMSNAP_APPLYTRACE table, specifically, any rows that have a value
   of WARNING for the OPERATION column. If any row is fetched, the
   DESCRIPTION column is included in the alert.

SUBSCRIPTIONS FAILING
   Specifies whether the Monitor program checks if processed subscription sets
   finished in error. These subscription set have rows in the
   IBMSNAP_APPLYTRAIL table with a value of -1 in the STATUS column.

SUBSCRIPTIONS DELAYED time
   Specifies whether the Monitor program checks if subscription sets were
   processed too late. The determination is based on the following formula:
   (LAST_RUN + user threshold in seconds > CURRENT TIMESTAMP).
SUBSCRIPTIONS INACTIVE
Specifies whether the Monitor program looks for subscription sets made inactive by the Apply program. Such sets are identified by a value of 0 for the ACTIVATE column and -1 for the STATUS column of the IBMSNAP_SUBS_SET table.

SUBSCRIPTIONS REFRESHED
Specifies whether the Monitor programs checks if a full refresh has been processed since the last Monitor cycle. See the FULL_REFRESH column in the IBMSNAP_APPLYTRAIL table for this information (rows from the IBMSNAP_APPLYTRAIL table whose values for FULL_REFRESH are ‘Y’). If any row is fetched, an alert is sent.

TRANSACTIONS REJECTED
Specifies that the Monitor program checks if any conflict has been detected by the Apply program when updating the source table and the replica tables. This check is valid only for subscriptions in an update-anywhere replication environment. See the IBMSNAP_APPLYTRAIL table for this information. If any row is fetched, an alert is sent.

REWORKED ROWS
Specifies whether the Monitor program checks if any rows were inserted into the IBMSNAP_APPLYTRAIL table since the last Monitor cycle for rows reworked in the target table. If the number of rows fetched exceeds the specified value, an alert is sent.

LATTENCY end-end-latency
Specifies whether the Monitor program checks if the total time required to process the data end-to-end (including time it took to capture it) is too high. If the value from the IBMSNAP_APPLYTRAIL table exceeds the specified value, an alert is sent.

Usage notes
• Specify the alert conditions in parentheses and separate them with commas.
• If you specify the same alert condition twice, the ASNCLP program issues an error.

Example
To alter an alert condition for the Apply program by removing the condition WARNINGS and no longer alerting the contact REPLADMIN when the condition occurs:
ALTER ALERT CONDITIONS FOR APPLY QUALIFIER MYAPPLY01 MONITOR QUALIFIER MONQUAL NOTIFY REPLADMIN (REMOVE WARNINGS)

ALTER ALERT CONDITIONS FOR CAPTURE command
Use the ALTER ALERT CONDITIONS FOR CAPTURE command to alter alert conditions for the Capture program.

Syntax

```
ALTER ALERT CONDITIONS FOR CAPTURE MONITOR-QUALIFIER mon-qual SCHEMA cap-schema
```
**Parameters**

**SCHEMA** *cap-schema*

Specifies the Capture schema for the server that you are monitoring. The default is ASN.

**MONITOR QUALIFIER** *mon-qual*

Specifies the Monitor qualifier.

**NOTIFY**

Specifies the contact or group of contacts to notify when the alert condition occurs.

- **CONTACT** *contact-name*
  
  Specifies the contact to notify.

- **GROUP** *group-name*
  
  Specifies the group to notify.

**OPERATOR CONSOLE**

This option is valid only if the monitor server is on a z/OS subsystem.

**ADD**

Specify to add an alert condition.
REMOVE
Specify to remove an alert condition.

CHANGE
Specify to change an alert condition.

STATUS DOWN
Specifies whether the Monitor program uses the asncmd status command to verify that the Capture program is running. The asncmd status command uses the DB2 Administration Server. If the Capture program is not running, an alert is sent.

STATUS LAST COMMIT time-secs
Specifies that the Monitor program calculates the difference between the values of the CURRENT TIMESTAMP and CURR_COMMIT_TIME columns of the IBMSNAP_RESTART table. This option has more delay than the STATUS DOWN option, but can be useful if you don’t run the DB2 Administration Server at the monitored server. If the calculated difference is greater than the number of seconds specified, an alert is sent.

ERRORS
Specifies that the Monitor program checks if any error messages were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of ERROR for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

WARNINGS
Specifies that the Monitor program checks if any warnings were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

CURRENT LATENCY latency
Specifies that the Monitor program calculates the current latency using the values of the CURR_COMMIT_TIME and MAX_COMMIT_TIME columns in the IBMSNAP_RESTART table. If the latency is greater than the number of seconds specified, an alert is sent.

HISTORIC LATENCY latency
Specifies that the Monitor program calculates the current latency using the values of the MONITOR_TIME and SYNCHTIME columns in the IBMSNAP_CAPMON table. If the latency is greater than the number of seconds specified, an alert is sent.

MEMORY memory
Specifies whether the Monitor program selects rows from the IBMSNAP_CAPMON table that were inserted since the last Monitor cycle to verify if the CURRENT_MEMORY column exceeded the specified value.

Usage notes
- Specify the alert conditions in parentheses and separate them with commas.
- If you specify the same alert condition twice, the ASNCLP program issues an error.

Example
To alter an alert condition for the Capture program by removing the condition MEMORY and no longer alerting the contact REPLADMIN when the condition occurs:
ALTER ALERT CONDITIONS FOR QAPPLY command

Use the ALTER ALERT CONDITIONS FOR QAPPLY command to alter alert conditions for the Q Apply program.

Syntax

```
ALTER ALERT CONDITIONS FOR QAPPLY SCHEMA schema MONITOR-QUALIFIER monitor-qualifier
```

```
notify-clause
```

```
add-or-remove-clause
```

```
change-clause
```

Parameters

**SCHEMA schema**

Specifies the Q Apply schema that qualifies the process to be monitored. The default is ASN.

**MONITOR QUALIFIER monitor-qualifier**

Specifies the monitor qualifier that groups the alert conditions:
ADD
Specify to add an alert condition.

REMOVE
Specify to remove an alert condition.

CHANGE
Specify to change an alert condition.

STATUS DOWN
Specifies that the Monitor program will use the asnqcml status command to verify if the Q Apply program is down.

ERRORS
Specifies that the Monitor program check if error messages were logged in the IBMQREP_APPLYTRACE table.

WARNINGS
Specifies that the Monitor program checks if any warnings were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

LATENCY seconds
Specifies that an alert will be sent when the difference in seconds of MONITOR_TIME and CURRENT_LOG_TIME in the IBMQREP_APPLYMON table exceeds the number of seconds specified.

EELATENCY seconds
Specifies that an alert will be sent when the value of the column END2END_LATENCY (in milliseconds) in the IBMQREP_APPLYMON table exceeds the number of milliseconds specified.

MEMORY megabytes
Specifies that the Monitor process will select rows from the IBMQREP_APPLYMON table that were inserted since the last Monitor cycle to verify if the CURRENT_MEMORY column exceeded the number of megabytes specified.

EXCEPTIONS
Specifies that an alert will be sent if there is any row in the IBMQREP_EXCEPTIONS table.

SPILL QUEUES DEPTH percentage
Specifies that the Monitor program will check whether the percentage of fullness of the spill queue is greater than specified percentage. The Monitor program checks this percentage only when any Q subscription is on the load state (the value of the STATE column in the IBMQREP_TARGETS table is L, D, F, or E).

QUEUE DEPTH queue_percent
Specifies that an alert will be sent when the specified percentage of the given queue is full.

RECEIVE QUEUES ALL INACTIVE
Specifies that an alert will be sent when the value of the STATE column in the IBMQREP_RECVQUEUES table changes to I (inactive) for any receive queue.

notify-clause:

CONTACT contact_name
Specifies the contact to notify when a defined alert condition is detected.
GROUP  group_name
    Specifies the group to notify when a defined alert condition is detected.

OPERATOR CONSOLE
    z/OS  Specifies that alert notifications are sent to the z/OS console. This option is valid only if the monitor server is on a z/OS subsystem.

Example

To alter an alert condition for the Q Apply program by removing the condition EXCEPTIONS and no longer alerting the contact REPLADMIN when the condition occurs:

ALTER ALERT CONDITIONS FOR QAPPLY MONITOR QUALIFIER MONQUAL
    NOTIFY REPLADMIN (REMOVE EXCEPTIONS)

ALTERN ALERT CONDITIONS FOR QCAPTURE command

Use the ALTER ALERT CONDITIONS FOR QCAPTURE command to alter the alert conditions for the Q Capture program.

Syntax

```plaintext
| ALTERN ALERT CONDITIONS FOR QCAPTURE | SCHEMA—schema | MONITOR-QUALIFIER—monitor-qualifier |
| notify-clause | ( | add-or-remove-clause | change-clause |
| notify-clause: | |
| NOTIFY | CONTACT—contact-name |
| GROUP—group-name |
| OPERATOR CONSOLE |

add-or-remove-clause: |

| ADD | REMOVE |
| STATUS DOWN |
| ERRORS |
| WARNINGS |
| LATENCY—seconds |
| MEMORY—megabytes |
| TRANSACTION SIZE—megabytes |
| SUBSCRIPTIONS INACTIVE |
| QUEUE DEPTH—queue-percent |

change-clause: |
Parameters

**SCHEMA** `schema`
- Specifies the Q Capture schema that qualifies the process to be monitored. The default is ASN.

**MONITOR QUALIFIER** `monitor-qualifier`
- Specifies the monitor qualifier that groups the alert conditions.

**ADD**
- Specify to add an alert condition.

**REMOVE**
- Specify to remove an alert condition.

**CHANGE**
- Specify to change an alert condition.

**STATUS DOWN**
- Specifies that the Monitor program will use the asnqccmd status command to verify if the Q Capture program is down.

**ERRORS**
- Specifies that the Monitor program check if error messages were logged in the IBMQREP_CAPTRACE table.

**WARNINGS**
- Specifies that the Monitor program checks if any warnings were logged in the IBMMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

**LATENCY** `seconds`
- Specifies that an alert will be sent when the difference in seconds of MONITOR_TIME and CURRENT_LOG_TIME in the IBMQREP_CAPMON table exceeds the number of seconds specified.

**MEMORY** `megabytes`
- Specifies that the Monitor process will select rows from the IBMQREP_CAPMON table that were inserted since the last Monitor cycle to verify if the CURRENT_MEMORY column exceeded the number of megabytes specified.

**TRANSACTION SIZE** `megabytes`
- Specifies that the Monitor process will select rows for the IBMMSNAP_CAPMON table to verify if any transaction size exceeded the number of megabytes specified.

**SUBSCRIPTIONS INACTIVE**
- Specifies that an alert will be sent when the value of the STATE column in the IBMQREP_SUBS table is I.
**QUEUE DEPTH** *queue-percent*

Specifies that an alert will be sent when the specified percentage of the given queue is full.

**notify-clause:**

**CONTACT** *contact_name*

Specifies the contact to notify when a defined alert condition is detected.

**GROUP** *group-name*

Specifies the group to notify when a defined alert condition is detected.

**OPERATOR** **CONSOLE**

Specifies that alert notifications are sent to the z/OS console. This option is valid only if the monitor server is on a z/OS subsystem.

**Example**

To alter an alert condition for the Q Capture program by removing the condition MEMORY and no longer alerting the contact REPLADMIN when the condition occurs:

```
ALTER ALERT CONDITIONS FOR QCAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
NOTIFY CONTACT REPLADMIN (REMOVE MEMORY 60)
```

**ALTER CONTACT command**

Use the **ALTER CONTACT** command to alter contact information, such as the contact name and mail address, that the Replication Alert Monitor program uses for notifications when a replication alert condition is detected.

**Syntax**

```
ALTER CONTACT contact-name EMAIL "email-address" PAGE "email-address" DESCRIPTION "description"
```

**Parameters**

**CONTACT** *contact-name*

Specifies the name of the contact. The contact must exist.

**EMAIL** *email-address*

Specifies the primary e-mail address for the contact. The double quotation marks are required.

**PAGE** *email-address*

Specifies the pager address for the contact. The double quotation marks are required.

**DESCRIPTION** *description*

Specifies a brief description for the contact. The double quotation marks are required.

**Example**

To alter a contact REPLADMIN by changing the e-mail address to repladmin@ibm.com:

```
ALTER CONTACT REPLADMIN EMAIL "repladmin@ibm.com"
```
**ALTER GROUP command**

Use the ALTER GROUP command to alter a group of replication monitor contacts.

**Syntax**

```
ALTER GROUP group-name
  DESCRIPTION "description"
  NEW CONTACTS contact-name1
  CONTACTS contact-name2
```

**Parameters**

- **group-name**
  Specifies the name of the group. The group must exist.

- **DESCRIPTION "description"**
  Specifies a brief description for the group. The double quotation marks are required.

- **NEW CONTACTS contact-name1**
  Specifies a comma-separated list of contacts that belong to this group. This list overwrites the existing list of contacts for the group.

- **CONTACTS contact-name2**

  - **ADD**
    Specifies a comma-separated list of contacts to add to this group.

  - **REMOVE**
    Specifies a comma-separated list of contacts to remove from this group.

**Example**

To alter a group MAINTENANCE by removing a contact PERFORMANCE:

```plaintext
ALTER GROUP MAINTENANCE CONTACTS PERFORMANCE REMOVE
```

**ALTER MONITOR SUSPENSION command**

Use the ALTER MONITOR SUSPENSION command to specify a different template for the monitor suspension, to change the start or end date for using the template, or to change the start or end date for suspending the monitor program if you do not use a template.

**Syntax**

```
ALTER MONITOR SUSPENSION name
  TEMPLATE template_name
  STARTING DATE date
```
Parameters

TEMPLATE
Specifies the template that you want to use for this suspension.

STARTING DATE
Specifies one of two different values, depending on whether you use a template for the suspension:

With template
Specifies the date that you want to start using the monitor suspension template.

Without template
Specifies the date on which the monitor program will be suspended. Use YYYY-MM-DD format.

ENDING DATE
Specifies one of two different values, depending on whether you use a template for the suspension:

With template
Specifies the date that you want to stop using the monitor suspension template.

Without template
Specifies the date when the monitor suspension ends. Use YYYY-MM-DD format.

Usage notes

To initiate the change, use the asnmcmd reinit command, or stop and start the monitor program.

Example 1

To change the suspension S1 so that it uses a different template, SATURDAY, and applies the template starting 2006-12-09:

```
ALTER MONITOR SUSPENSION NAME S1 TEMPLATE SATURDAY STARTING DATE 2006-12-09
```

Example 2

To change the suspension S2 so that it uses a template, LUNCH1, starting 2007-01-01 and ending 2007-06-30:

```
ALTER MONITOR SUSPENSION NAME S2 TEMPLATE LUNCH1 STARTING DATE 2007-01-01 ENDING DATE 2007-06-30
```

ALTER MONITOR SUSPENSION TEMPLATE command

Use the ALTER MONITOR SUSPENSION TEMPLATE command to change the frequency and duration of periods that the monitor program is suspended.
Syntax

ALTER MONITOR SUSPENSION TEMPLATE template_name START TIME HH:MM:SS

REPEATS occurrence-clause

occurrence-clause:

- DAILY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK SUNDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK MONDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK TUESDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK WEDNESDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK THURSDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK FRIDAY FOR DURATION n HOURS
- WEEKLY DAY OF WEEK SATURDAY FOR DURATION n HOURS

Parameters

START TIME
   Specifies the time at which the monitor program will be suspended. Use HH:MM:SS format. The default value is 00:00:00.

REPEATS
   Specifies which days the monitor program will be suspended and for how long.

Usage notes

To initiate the change, use the asnmcmd reinit command, or stop and start the monitor program.

Example 1

To change a template so that it suspends the monitor program from 00:00:00 to 03:00:00 every SUNDAY for one year:
ALTER MONITOR SUSPENSION TEMPLATE sunday START TIME 00:00:00 REPEATS WEEKLY DAY OF WEEK SUNDAY FOR DURATION 3 HOURS

Example 2

To lengthen a template that suspends the monitor program during the lunch hour every day to 90 minutes:
ALTER MONITOR SUSPENSION TEMPLATE lunch START TIME 12:00:00 REPEATS DAILY FOR DURATION 90 MINUTES

CREATE ALERT CONDITIONS FOR APPLY command

Use the CREATE ALERT CONDITIONS FOR APPLY command to create alert conditions for the Apply program. Each entry represents a condition that the Replication Alert Monitor program looks for. If the condition is true, the Monitor program sends an alert to the corresponding contact or group, or to the operator console.
Syntax

```plaintext
CREATE ALERT CONDITIONS FOR APPLY_QUALIFIER qual-name
  SET NAME set-name

MONITOR-QUALIFIER mon-qual NOTIFY
  CONTACT contact-name
  GROUP group-name
  OPERATOR CONSOLE

(STATUS DOWN
  ERRORS
  WARNINGS
  SUBSCRIPTIONS FAILING
  SUBSCRIPTIONS DELAYED time
  SUBSCRIPTIONS INACTIVE
  SUBSCRIPTIONS REFRESHED
  TRANSACTIONS REJECTED
  REWORKED ROWS rows
  LATENCY end-end-latency)
```

Parameters

**APPLY QUALIFIER** `qual-name`
Specifies the Apply qualifier.

**SET NAME** `set-name`
Specifies the subscription set name. If you do not specify a subscription set name, all of the set names in the Apply qualifier will be assumed.

**MONITOR QUALIFIER** `mon-qual`
Specifies the Monitor qualifier.

**NOTIFY**
Specifies the contact or group of contacts to notify when the alert condition occurs.

  **CONTACT** `contact-name`
  Specifies the contact to notify.

  **GROUP** `group-name`
  Specifies the group to notify.

**OPERATOR CONSOLE**
Specifies that alert notifications are sent to the z/OS console. This option is valid only if the monitor server is on a z/OS subsystem.

**STATUS DOWN**
Specifies whether the Monitor program uses the asncmd status command to verify that the Apply program is running. The asncmd status command uses the DB2 Administration Server for non-OS/400 systems. If the Apply program is not running, an alert is sent.

**ERRORS**
Specifies that the Monitor program checks if any error messages were logged in the IBMSNAP_APPLYTRACE table, specifically, any rows that have a value of ERROR for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.
WARNINGS
Specifies that the Monitor program checks if any warnings were logged in the
IBMSNAP_APPLYTRACE table, specifically, any rows that have a value of
WARNING for the OPERATION column. If any row is fetched, the
DESCRIPTION column is included in the alert.

SUBSCRIPTIONS FAILING
Specifies whether the Monitor program checks if processed subscription sets finished in error. These subscription set have rows in the
IBMSNAP_APPLYTRAIL table with a value of -1 in the STATUS column.

SUBSCRIPTIONS DELAYED time
Specifies whether the Monitor program checks if subscription sets were processed too late. The determination is based on the following formula:
(LAST_RUN + user threshold in seconds > CURRENT TIMESTAMP).

SUBSCRIPTIONS INACTIVE
Specifies whether the Monitor program looks for subscription sets made inactive by the Apply program. Such sets are identified by a value of 0 for the ACTIVATE column and -1 for the STATUS column of the IBMSNAP_SUBS_SET table.

SUBSCRIPTIONS REFRESHED
Specifies whether the Monitor program checks if a full refresh has been processed since the last Monitor cycle. See the FULL_REFRESH column in the IBMSNAP_APPLYTRAIL table for this information (rows from the IBMSNAP_APPLYTRAIL table whose values for FULL_REFRESH are ‘Y’). If any row is fetched, an alert is sent.

TRANSACTIONS REJECTED
Specifies that the Monitor program checks if any conflict has been detected by the Apply program when updating the source table and the replica tables. This check is valid only for subscriptions in an update-anywhere replication environment. See the IBMSNAP_APPLYTRAIL table for this information. If any row is fetched, an alert is sent.

REWORKED ROWS rows
Specifies whether the Monitor program checks if any rows were inserted into the IBMSNAP_APPLYTRAIL table since the last Monitor cycle for rows reworked in the target table. If the number of rows fetched exceeds the specified value, an alert is sent.

LATENCY end-end-latency
Specifies whether the Monitor program checks if the total time required to process the data end-to-end (including time it took to capture it) is too high. If the value from the IBMSNAP_APPLYTRAIL table exceeds the specified value, an alert is sent.

Usage notes
• Specify the alert conditions in parentheses and separate them with commas.
• If you specify the same alert condition twice, the ASNCLP program issues an error.

Example
To create alert conditions for the Apply program that sends an alert to the contact REPLADMIN when a condition occurs:
CREATE ALERT CONDITIONS FOR APPLY QUALIFIER MYAPPLY01 MONITOR QUALIFIER MONQUAL
NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS, SUBSCRIPTIONS FAILING,
SUBSCRIPTIONS DELAYED 300, SUBSCRIPTIONS INACTIVE, SUBSCRIPTIONS REFRESHED,
TRANSACTIONS REJECTED, REWORKED ROWS 2, LATENCY 360)

CREATE ALERT CONDITIONS FOR CAPTURE command

Use the CREATE ALERT CONDITIONS FOR CAPTURE command to create alert conditions for the Capture program. Each entry represents a condition that the Replication Alert Monitor program looks for. If the condition is true, the Monitor program sends an alert to the corresponding contact or group, or to the operator console.

Syntax

```plaintext
CREATE ALERT CONDITIONS FOR CAPTURE
SCHEMA cap-schema MONITOR QUALIFIER mon-qual NOTIFICATION
CONTACT contact-name GROUP group-name OPERATOR CONSOLE
```

Parameters

SCHEMA cap-schema
- Specifies the Capture schema for the server that you are monitoring. The default is ASN.

MONITOR QUALIFIER mon-qual
- Specifies the Monitor qualifier.

NOTIFY
- Specifies the contact or group of contacts to notify when the alert condition occurs.

CONTACT contact-name
- Specifies the contact to notify.

GROUP group-name
- Specifies the group to notify.

OPERATOR CONSOLE
- Specifies that alert notifications are sent to the z/OS console. This option is valid only if the monitor server is on a z/OS subsystem.

STATUS DOWN
- Specifies whether the Monitor program uses the asnccmd status command to verify that the Capture program is running. The asnccmd status command uses the DB2 Administration Server. If the Capture program is not running, an alert is sent.
STATUS LAST COMMIT time-secs
Specifies that the Monitor program calculates the difference between the values of the CURRENT TIMESTAMP and CURR_COMMIT_TIME columns of the IBMSNAP_RESTART table. This option has more delay than the STATUS DOWN option, but can be useful if you do not run the DB2 Administration Server at the monitored server. If the calculated difference is greater than the number of seconds specified, an alert is sent.

ERRORS
Specifies that the Monitor program checks if any error messages were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of ERROR for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

WARNINGS
Specifies that the Monitor program checks if any warnings were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

CURRENT LATENCY latency
Specifies that the Monitor program calculates the current latency using the values of the CURR_COMMIT_TIME and MAX_COMMIT_TIME columns in the IBMSNAP_RESTART table. If the latency is greater than the number of seconds specified, an alert is sent.

HISTORIC LATENCY latency
Specifies that the Monitor program calculates the current latency using the values of the MONITOR_TIME and SYNCHTIME columns in the IBMSNAP_CAPMON table. If the latency is greater than the number of seconds specified, an alert is sent.

MEMORY memory
Specifies whether the Monitor program selects rows from the IBMSNAP_CAPMON table that were inserted since the last Monitor cycle to verify if the CURRENT_MEMORY column exceeded the specified value.

Usage notes
If you specify the same alert condition twice, the ASNCLP program issues an error.

Example
To create alert conditions for the Capture program that sends an alert to the contact REPLADMIN when a condition occurs:

CREATE ALERT CONDITIONS FOR CAPTURE QUALIFIER MYAPPLY01 MONITOR QUALIFIER MONQUAL NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS, SUBSCRIPTION FAILING, SUBSCRIPTION DELAYED 300, SUBSCRIPTIONS INACTIVE, SUBSCRIPTIONS REFRESHED, TRANSACTION REJECTED, REWORKED ROWS 2, LATENCY 360)

CREATE ALERT CONDITIONS FOR QAPPLY command
Use the CREATE ALERT CONDITIONS FOR QAPPLY command to create alert conditions for the Q Apply program. Each entry represents a condition that the Replication Alert Monitor program looks for. If the condition is true, the Monitor program sends an alert to the corresponding contact or group, or to the operator console.
Syntax

```
CREATE ALERT CONDITIONS FOR QAPPLY
  SCHEMA schema
  MONITOR-QUALIFIER monitor-qualifier
  notification-list-definition
```

```
notification-list-definition:
[ ] (  
  STATUS DOWN
  ERRORS
  WARNINGS
  LATENCY--seconds
  EELATENCY--seconds
  MEMORY--megabytes
  EXCEPTIONS
  --SPILL QUEUES DEPTH--percentage--
  QUEUE DEPTH--queue-percent
  RECEIVE QUEUES ALL INACTIVE
)
```

Parameters

**SCHEMA schema**
Specifies the Q Apply schema that qualifies the process to be monitored. The default is ASN.

**MONITOR QUALIFIER monitor-qualifier**
Specifies the monitor qualifier that groups the alert conditions:

**STATUS DOWN**
Specifies that the Monitor program will use the asnacmd status command to verify if the Q Apply program is down.

**ERRORS**
Specifies that the Monitor program checks if error messages were logged in the IBMQREP_APPLYTRACE table.

**WARNINGS**
Specifies that the Monitor program checks if any warnings were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

**LATENCY seconds**
Specifies that an alert will be sent when the difference in seconds of MONITOR_TIME and CURRENT_LOG_TIME in the IBMQREP_APPLYMON table exceeds the number of seconds specified.

**EELATENCY seconds**
Specifies that an alert will be sent when the value of the column END2END_LATENCY (in milliseconds) in the IBMQREP_APPLYMON table exceeds the number of milliseconds specified.
MEMORY megabytes
Specifications: the Monitor process will select rows from the
IBMQREP_APPLYMON table that were inserted since the last Monitor cycle to
verify if the CURRENT_MEMORY column exceeded the number of megabytes
specified.

EXCEPTIONS
Specifications: that an alert will be sent if there is any row in the
IBMQREP_EXCEPTIONS table.

SPILL QUEUES DEPTH percentage
Specifications: that the Monitor program will check whether the percentage of
fullness of the spill queue is greater than specified percentage. The Monitor
program checks this percentage only when any Q subscription is on the load
state (the value of the STATE column in the IBMQREP_TARGETS table is L, D,
F, or E).

QUEUE DEPTH queue-percent
Specifications: that an alert will be sent when the specified percentage of the given
queue is full.

RECEIVE QUEUES ALL INACTIVE
Specifications: that an alert will be sent when the value of the STATE column in the
IBMQREP_RECVQUEUES table changes to I (inactive) for any receive queue.

notification-list-definition:

CONTACT contact_name
Specifications: the contact to notify when a defined alert condition is detected.

GROUP group-name
Specifications: the group to notify when a defined alert condition is detected.

OPERATOR CONSOLE
   z/OS Specifies that alert notifications are sent to the z/OS
   console. This option is valid only if the monitor server is on a z/OS subsystem.

Example
To create alert conditions for the Q Apply program that send an alert to the contact
REPLADMIN when a condition occurs:

CREATE CONDITIONS FOR QAPPLY MONITOR QUALIFIER MONQUAL
NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS,
LATENCY 360, EXCEPTIONS)

CREATE ALERT CONDITIONS FOR QCAPTURE command
Use the CREATE ALERT CONDITIONS FOR QCAPTURE command to create alert
conditions for the Q Capture program. Each entry represents a condition that the
Replication Alert Monitor program looks for. If the condition is true, the Monitor
program sends an alert to the corresponding contact or group, or to the operator
console.

Syntax

CREATE ALERT CONDITIONS FOR QCAPTURE SCHEMA schema MONITOR QUALIFIER monitor-qualifier
notification-list-definition:

- NOTIFY
- CONTACT contact-name
- GROUP group-name
- OPERATOR CONSOLE

Parameters

**SCHEMA schema**
Specifies the Q Capture schema that qualifies the process to be monitored. The default is ASN.

**MONITOR QUALIFIER monitor-qualifier**
Specifies the monitor qualifier that groups the alert conditions.

**STATUS DOWN**
Specifies that the Monitor program will use the asnqccmd status command to verify if the Q Capture program is down.

**ERRORS**
Specifies that the Monitor program checks if error messages were logged in the IBMQREP_CAPTRACE table.

**WARNINGS**
Specifies that the Monitor program checks if any warnings were logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of WARNING for the OPERATION column. If any row is fetched, the DESCRIPTION column is included in the alert.

**LATENCY seconds**
Specifies that an alert will be sent when the difference in seconds of MONITOR_TIME and CURRENT_LOG_TIME in the IBMQREP_CAPMON table exceeds the number of seconds specified.

**MEMORY megabytes**
Specifies that the Monitor process will select rows from the IBMQREP_CAPMON table that were inserted since the last Monitor cycle to verify if the CURRENT_MEMORY column exceeded the number of megabytes specified.

**TRANSACTION SIZE megabytes**
Specifies that the Monitor process will select rows for the IBMSNAP_CAPMON table to verify if any transaction size exceeded the number of megabytes specified.

**SUBSCRIPTIONS INACTIVE**
Specifies that an alert will be sent when the value of the STATE column in the IBMQREP_SUBS table is I.
QUEUE DEPTH queue-percent
   Specifies that an alert will be sent when the specified percentage of the given queue is full.

notification-list-definition:
CONTACT contact_name
   Specifies the contact to notify when a defined alert condition is detected.

GROUP group-name
   Specifies the group to notify when a defined alert condition is detected.

OPERATOR CONSOLE
   Specifies that alert notifications are sent to the z/OS console. This option is valid only if the monitor server is on a z/OS subsystem.

Example
To create alert conditions for the Q Capture program that sends an alert to the contact REPLADMIN when a condition occurs:
CREATE ALERT CONDITIONS FOR QCAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS, LATENCY 30, MEMORY 60)

CREATE CONTACT command
Use the CREATE CONTACT command to create contact information, such as the contact name and e-mail address, that the Replication Alert Monitor program uses for notifications when a replication alert condition is detected. You can optionally associate a contact to a pre-existing group.

Syntax
```
CREATE CONTACT contact-name [GROUP group-name] EMAIL "email-address" [PAGE "email-address"] [DESCRIPTION "description"]
```

Parameters
CONTACT contact-name
   Specifies the name of the contact. This name cannot match another contact already defined.

GROUP group-name
   Specifies the name of the group to add the contact to. The group must be already defined.

EMAIL "email-address"
   Specifies the primary e-mail address for the contact. The double quotation marks are required.

PAGE "email-address"
   Specifies the pager address for the contact. The double quotation marks are required.
DESCRIPTION "description"
   Specifies a brief description for the contact. The double quotation marks are
   required.

Example

To create a contact REPLADMIN with an e-mail address repladmin@us.ibm.com:
CREATE CONTACT REPLADMIN EMAIL "repladmin@us.ibm.com"
DESCRIPTION "replication administration"

CREATE CONTROL TABLES FOR command

Use the CREATE CONTROL TABLES FOR command to create a new set of
Replication Alert Monitor control tables.

Syntax

```
CREATE CONTROL TABLES FOR MONITOR CONTROL SERVER IN ZOS zos-ts-clause

zos-ts-clause:

   ALERTS DB dbname tsname prof-clause
      NAMING PREFIX prefix

   PAGE LOCK DB dbname tsname prof-clause
      NAMING PREFIX prefix

   ROW LOCK DB dbname tsname prof-clause
      NAMING PREFIX prefix

uw-ts-clause:

   OTHERS tsname prof-clause
      NAMING PREFIX prefix

fed-ts-clause:

   OTHERS tsname prof-clause
      SCHEMA schemaname
      NAMING PREFIX prefix

prof-clause:

   CREATE USING PROFILE pname
   REUSE
```

180   ASNCLP Program Reference for Replication and Event Publishing
Parameters

**MONITOR CONTROL SERVER**
Specify to create replication control tables for the Monitor control server.

**IN**
Specifies the table space. If you do not specify the **IN** clause, the CREATE CONTROL TABLES command uses the DB2 defaults for table spaces.

**ZOS**
Specifies z/OS or OS/390.

**UW**
Specifies UNIX or Windows.

**NONIBM**
Specifies non-DB2 data sources.

**ALERTS**
Specifies an existing database on z/OS to create the control tables in. This keyword is valid only when creating monitor control servers.

**PAGE LOCK**
Specifies the table space for replication control tables that require page-level locking. The table must be in an existing database.

**ROW LOCK**
Specifies the table space for replication control tables that require row-level locking. The table must be in an existing database.

**DB dbname**
Specifies the name of an existing database. You must specify the database name, even if you set the database name in the profile. This command does not create the database.

**OTHERS**
Specifies the table space for all replication control tables except the UOW table.

**tsname**
Specifies the table space name for the monitor alerts table. The *tsname* input can be a heterogeneous segment or table space name.

**NAMING PREFIX prefix**
Specifies a naming prefix for the control tables.

**SCHEMA schemaname**
Specifies the remote schema name for heterogeneous replication. The default is the remote user ID. For non-DB2 databases, you can specify a table space name or a segment name for those remote sources that support them.

**CREATE USING PROFILE pname**
Specify to create the control tables using the *pname* profile. If you specify the CREATE USING PROFILE parameter, the ASNCLP program uses *tsname* as the key (For z/OS, the key is *dbname.tsname*).

**REUSE**
Specify to reuse the current DDL object. You must issue the CREATE USING PROFILE parameter before you can use the **REUSE** parameter. When you specify the **REUSE** parameter, the ASNCLP program checks if the DDL object exists for the *tsname*:

- If the DDL object exists, the ASNCLP program resets the flags and passes the fully populated DDL.
If the DDL object does not exist, the ASNCLP program displays a syntax error saying that the `CREATE USING PROFILE` parameter is expected.

Example 1

To create the Monitor control tables:

`CREATE CONTROL TABLES FOR MONITOR CONTROL SERVER`

**CREATE GROUP command**

The `CREATE GROUP` command creates a group of replication monitor contacts.

**Syntax**

```
CREATE GROUP group-name DESCRIPTION "description" CONTACTS contact-name
```

**Parameters**

`group-name`
- Specifies the name of the group. This name cannot match another group already defined. This parameter is required.

`DESCRIPTION "description"`
- Specifies a brief description for the group. The double quotation marks are required.

`CONTACTS contact-name`
- Specifies a comma-separated list of contacts that belong to this group.

**Example**

To create a group `MAINTENANCE` that contains contacts `REPLADMIN` and `PERFORMANCE`:

`CREATE GROUP MAINTENANCE CONTACTS REPLADMIN, PERFORMANCE`

**CREATE MONITOR SUSPENSION command**

Use the `CREATE MONITOR SUSPENSION` command to suspend the monitor program. You can specify a start and end date or use a template that defines a repeating pattern of suspensions.

**Syntax**

```
CREATE MONITOR SUSPENSION name FOR SERVER server_name ALIAS server_alias STARTING DATE date
USING TEMPLATE template_name STARTING TIME starting_time ENDING DATE date ENDING TIME ending_time
```

```
```
Parameters

SERVER
Specifies the name of the DB2 database where you want to suspend the monitor program.

z/OS: This value represents the DB2 subsystem location name.

ALIAS
Linux, UNIX, Windows: The DB2 alias for the database where you want to suspend the monitor program.

STARTING DATE
Specifies one of two different values, depending on whether you use a template for the suspension:

With template
Specifies the date that you want to start using the monitor suspension template.

Without template
Specifies the date on which the monitor program will be suspended. Use YYYY-MM-DD format.

USING TEMPLATE
Specifies that you want to use a template to set the start time and other characteristics of the suspension. You define the template by using the CREATE MONITOR SUSPENSION TEMPLATE command.

STARTING TIME
Specifies the time when the monitor suspension begins. Use HH:MM:SS format. The default is 00:00:00.

ENDING DATE
Specifies one of two different values, depending on whether you use a template for the suspension:

With template
Specifies the date that you want to stop using the monitor suspension template.

Without template
Specifies the date when the monitor suspension ends. Use YYYY-MM-DD format.

ENDING TIME
Specifies one of two different values, depending on whether you use a template for the suspension:

With template
Specifies the time that you want to stop using the monitor suspension template.

Without template
Specifies the time when the monitor suspension ends.

Use HH:MM:SS format for the ending time. The default is 00:00:00.

Example 1
To create a suspension S1 on the monitored server QSRVR1 that uses the template SUNDAY:
CREATE MONITOR SUSPENSION NAME S1 FOR SERVER QSRVR1 STARTING DATE 2006-12-10 USING TEMPLATE SUNDAY ENDING DATE 2007-12-31

Example 2

To create a suspension S2 on the monitored server QSRVR2 that does not use a template but suspends the monitor during the month of December:

CREATE MONITOR SUSPENSION NAME S2 FOR SERVER QSRVR2 STARTING DATE 2006-11-30 STARTING TIME 00:00:00 ENDING DATE 2006-12-31 ENDING TIME 24:00:00

CREATE MONITOR SUSPENSION TEMPLATE command

Use the CREATE MONITOR SUSPENSION TEMPLATE command to define the frequency and duration of periods that the monitor program is suspended.

Syntax

```
CREATE MONITOR SUSPENSION TEMPLATE <template_name>
  START TIME <HH:MM:SS>
  REPEATS <occurrence-clause>
```

occurrence-clause:

```
  DAILY FOR DURATION <n> HOURS
  WEEKLY DAY OF WEEK <SUNDAY> FOR DURATION <n> HOURS
       <MINUTES DAYS>
```

Parameters

START TIME

Specifies the time at which the monitor program will be suspended, in HH:MM:SS (hours:minutes:seconds) format. The default value is 00:00:00.

REPEATS

Specifies which days the monitor program will be suspended, and for how long.

Example 1

To create a template that suspends the monitor program from 00:00:00 to 04:00:00 every Sunday:

```
CREATE MONITOR SUSPENSION TEMPLATE SUNDAY START TIME 00:00:00 REPEATS WEEKLY
       DAY OF WEEK SUNDAY FOR DURATION 4 HOURS
```

Example 2

To create a template that suspends the monitor program during the lunch hour every day:

```
CREATE MONITOR SUSPENSION TEMPLATE LUNCH START TIME 11:30:00 FOR DURATION 1 HOUR
```

Example 3

To create a template that suspends the monitor program every Friday, starting at 10:00:00 for 4 hours:

```
CREATE MONITOR SUSPENSION TEMPLATE FRIDAY START TIME 10:00:00 REPEATS FOR DURATION 4 HOURS
```

Example 4

To create a template that suspends the monitor program every Monday, starting at 08:00:00 for 2 hours:

```
CREATE MONITOR SUSPENSION TEMPLATE MONDAY START TIME 08:00:00 REPEATS WEEKLY FOR DURATION 2 HOURS
```

Example 5

To create a template that suspends the monitor program every Saturday, starting at 09:00:00 for 3 hours:

```
CREATE MONITOR SUSPENSION TEMPLATE SATURDAY START TIME 09:00:00 REPEATS WEEKLY FOR DURATION 3 HOURS
```
DELEGATE CONTACT command

Use the DELEGATE CONTACT command to delegate an existing contact to a new contact for a specific period of time.

Syntax

```plaintext
>>DELEGATE CONTACT contact-name1 TO contact-name2 FROM "start-date" TO "end-date" <<
```

Parameters

- **CONTACT contact-name1**: Specifies the name of the contact to be delegated. The contact must exist.
- **TO contact-name2**: Specifies the new contact for all alert conditions (if any) that refer to the contact being delegated. The contact must exist.
- **FROM "start-date"**: Specifies the date when the delegation starts. The date is sensitive to the DB2 locale. The double quotation marks are required.
- **TO "end-date"**: Specifies the date when the delegation ends. The date is sensitive to the DB2 locale. The double quotation marks are required.

Example

To delegate alerts from one (REPLADMIN) contact to another (PERFORMACE) for a given period of time:

```plaintext
DELEGATE CONTACT REPLADMIN TO PERFORMACE FROM "2007-11-22" TO "2007-12-06"
```

DROP ALERT CONDITIONS FOR APPLY command

Use the DROP ALERT CONDITIONS FOR APPLY command to drop alert conditions for the Apply program.

Syntax

```plaintext
>>DROP ALERT CONDITIONS FOR APPLY QUALIFIER apply-qual MONITOR QUALIFIER mon-qual <<
```

Parameters

- **APPLY QUALIFIER apply-qual**: Specifies the Apply qualifier.
- **MONITOR QUALIFIER mon-qual**: Specifies the Monitor qualifier.

Example

To drop alert conditions for the Apply program:

```plaintext
DROP ALERT CONDITIONS FOR APPLY QUALIFIER MYAPPLY01 MONITOR QUALIFIER MONQUAL
```
DROP ALERT CONDITIONS FOR CAPTURE command

Use the DROP ALERT CONDITIONS FOR CAPTURE command to drop alert conditions for the Capture program.

Syntax

```
DROP ALERT CONDITIONS FOR CAPTURE SCHEMA cap-schema MONITOR QUALIFIER mon-qual
```

Parameters

- **SCHEMA cap-schema**
  - Specifies the Capture schema for the server that you are monitoring.

- **MONITOR QUALIFIER mon-qual**
  - Specifies the Monitor qualifier.

Example

To drop alert conditions for the Capture program:

```
DROP ALERT CONDITIONS FOR CAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
```

DROP ALERT CONDITIONS FOR QAPPLY command

Use the DROP ALERT CONDITIONS FOR QAPPLY command to drop alert conditions for the Q Apply program.

Syntax

```
DROP ALERT CONDITIONS FOR QAPPLY SCHEMA schema MONITOR QUALIFIER monitor-qualifier
```

Parameters

- **SCHEMA schema**
  - Specifies the Q Apply schema that qualifies the process to be monitored.

- **MONITOR QUALIFIER monitor-qualifier**
  - Specifies the monitor qualifier grouping the alert conditions.

Example

To drop alert conditions for the Q Apply program:

```
DROP ALERT CONDITIONS FOR QAPPLY SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
```

DROP ALERT CONDITIONS FOR QCAPTURE command

Use the DROP ALERT CONDITIONS FOR QCAPTURE command to drop alert conditions for the Q Capture program.

Syntax

```
DROP ALERT CONDITIONS FOR QCAPTURE SCHEMA schema MONITOR QUALIFIER monitor-qualifier
```
Parameters

**SCHEMA schema**
Specifies the Q Capture schema that qualifies the process to be monitored.

**MONITOR QUALIFIER monitor-qualifier**
Specifies the monitor qualifier that groups the alert conditions.

Example

To drop alert conditions for the Q Capture program:

```
DROP ALERT CONDITIONS FOR QCAPTURE SCHEMA ASN1 MONITOR QUALIFIER MONQUAL
```

**DROP CONTACT command**

Use the DROP CONTACT command to drop an existing contact.

**Syntax**

```
DROP CONTACT contact-name1 [SUBSTITUTE WITH contact-name2]
```

**Parameters**

**CONTACT contact-name1**
Specifies the name of the contact. The contact must exist.

**SUBSTITUTE WITH contact-name2**
Specifies the name of a contact. The contact must exist. If the contact being deleted is referenced by any alert conditions, then the alert conditions will now reference the contact represented in this clause.

**Usage notes**

If you drop a contact that is the only one referred by an alert condition, this command returns an error. In this case, you must either delete the alert condition before you drop the contact, or use the SUBSTITUTE WITH clause.

**Example**

To drop a contact REPLADMIN:

```
DROP CONTACT REPLADMIN
```

**DROP GROUP command**

Use the DROP GROUP command to drop a group of replication monitor contacts.

**Syntax**

```
DROP GROUP group-name
```

**Parameters**

**group-name**
Specifies the name of the group. The group must exist.
Usage notes
If you drop a group that is the only one referred to by an alert condition, and there are no individual contacts referred to by the alert condition, this command returns an error.

Example
To drop a group MAINTENANCE:
DROP GROUP MAINTENANCE

DROP MONITOR SUSPENSION command
Use the DROP MONITOR SUSPENSION command to delete a suspension from the monitor control tables.

Syntax
```
DROP MONITOR SUSPENSION—name
```

Parameters

name
Specifies the template that you want to delete.

Usage notes
After you remove the suspension, reinitialize the monitor or stop and start the monitor to prompt it to read its control tables and end the suspension.

Example
To delete the suspension S1:
DROP MONITOR SUSPENSION NAME S1

DROP MONITOR SUSPENSION TEMPLATE command
Use the DROP MONITOR SUSPENSION TEMPLATE command to delete a template from the monitor control tables.

Syntax
```
DROP MONITOR SUSPENSION TEMPLATE—template_name
```

Parameters

template_name
Specifies the name of an existing template.

Example
To drop the template named that is named sunday:
DROP MONITOR SUSPENSION TEMPLATE sunday
LIST MONITOR SUSPENSION command

Use the LIST MONITOR SUSPENSION command to generate a list of suspensions that are defined on a monitor control server. The command sends a report that shows the suspension name and other properties to the standard output (stdout).

Syntax

```
LIST MONITOR SUSPENSION
```

LIST MONITOR SUSPENSION TEMPLATE command

Use the LIST MONITOR SUSPENSION TEMPLATE command to generate a list of suspension templates on a monitor control server. The command sends a report that shows the template name and other properties to the standard output (stdout).

Syntax

```
LIST MONITOR SUSPENSION TEMPLATE
```

Example

The following example shows the output of the LIST MONITOR SUSPENSION TEMPLATE command:

<table>
<thead>
<tr>
<th>TEMPLATE_NAME</th>
<th>START_TIME</th>
<th>FREQUENCY</th>
<th>DURATION</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>daytemp1</td>
<td>12:00:00</td>
<td>DAILY</td>
<td>4</td>
<td>HOURS</td>
</tr>
<tr>
<td>wednesdaytemp2</td>
<td>00:00:00</td>
<td>WEDNESDAY</td>
<td>2</td>
<td>DAYS</td>
</tr>
<tr>
<td>minutestemp3</td>
<td>17:30:00</td>
<td>SUNDAY</td>
<td>30</td>
<td>MINUTES</td>
</tr>
</tbody>
</table>

SET OUTPUT command

Use the SET OUTPUT command to define output files for the ASNCLP program. The output files contain the SQL statements needed to set up Q replication and event publishing.

Syntax

```
SET OUTPUT MONITOR SCRIPT "monfname"
```

Parameters

MONITOR SCRIPT "monfname"

Specifies the output file name for scripts that run at the Monitor control server. The default file name is replmonitor.sql.

Usage notes

- If a script already exists, the new script appends to the current script.
- The double quotation marks in the command syntax are required.
Example 1

To name the monitor script output file "monitor.sql":

```
SET OUTPUT MONITOR SCRIPT "monitor.sql"
```

SET SERVER command

Use the SET SERVER command to specify the server (database) used in the
ASNCLP session. You can specify authentication information and other required
parameters for connecting to the server.

You should always set the Monitor control server before running the monitor
administration commands.

Syntax

```
SET SERVER MONITOR TO NULLS
  |     DB dbalias
  |     DBALIAS aliasname
  |     DBNAME dbname
  | other-options
```

other-options:

```
| ID userid
| PASSWORD pwd
```

Parameters

**MONITOR**

Specify to set the database as a monitor server.

**NULLS**

Specify to set the server name to NULLS. This option resets a previously set
server name.

**DB dbalias**

Specifies the database alias name.

**DBALIAS aliasname**

Specifies the database alias name.

**DBNAME dbname**

Specifies the database name.

**ID userid**

Specifies the user ID to use to connect to the database.

**PASSWORD pwd**

Specifies the password to use to connect to the database. If you specify the
user ID and do not specify the password, you will be prompted to enter the
password. The password is hidden as you type.

Example

To set the monitor server to the SAMPLE database:

```
SET SERVER MONITOR TO DB SAMPLE
```
**SUBSTITUTE CONTACT command**

Use the SUBSTITUTE CONTACT command to substitute one existing contact with another existing contact.

**Syntax**

```
SUBSTITUTE CONTACT contact-name1 WITH contact-name2
```

**Parameters**

- `contact-name1`  
  Specifies the name of the contact to be substituted. The contact must exist.

- `WITH contact-name2`  
  Specifies the new contact for all alert conditions (if any) that refer to the contact being substituted. The contact must exist.

**Example**

To substitute one contact (REPLADMIN) for another (PERFORMACE):

```
SUBSTITUTE CONTACT REPLADMIN WITH PERFORMACE
```
Accessing information about the product

IBM® has several methods for you to learn about products and services.

You can find the latest information on the Web:


To access product documentation, go to publib.boulder.ibm.com/infocenter/db2luw/v9r5/topic/

You can order IBM publications online or through your local IBM representative.

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To order publications by telephone in the United States, call 1-800-879-2755.

To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

Providing comments on the documentation

Please send any comments that you have about this information or other documentation.

Your feedback helps IBM to provide quality information. You can use any of the following methods to provide comments:

- Send your comments using the online readers’ comment form at www.ibm.com/software/awdtools/rcf/
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the information (if applicable). If you are commenting on specific text, please include the location of the text (for example, a title, a table number, or a page number).
Accessible documentation

Documentation is provided in XHTML format, which is viewable in most Web browsers.

XHTML allows you to view documentation according to the display preferences that you set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen reader.
Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.
Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.
Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**Trademarks**

IBM trademarks and certain non-IBM trademarks are marked at their first occurrence in this document.


The following terms are trademarks or registered trademarks of other companies:

Adobe®, the Adobe logo, PostScript®, the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine™ is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel®, Intel logo, Intel Inside® logo, Intel Centrino®, Intel Centrino logo, Celeron®, Intel Xeon®, Intel SpeedStep®, Itanium® and Pentium® are trademarks of Intel Corporation in the United States, other countries, or both.

Java™ and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT® and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

ITIL® is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library® is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Other company, product or service names may be trademarks or service marks of others.
Index

A
accessibility 193, 195
ALTER ALERT CONDITIONS FOR APPLY command 159
ALTER ALERT CONDITIONS FOR CAPTURE command 161
ALTER ALERT CONDITIONS FOR QAPPLY command 164
ALTER ALERT CONDITIONS FOR QCAPTURE command 166
ALTER CAPTURE PARAMETERS command (Q replication) 84
ALTER CONFIGURATION APPLY command 85
ALTER CONTACT command 168
ALTER GROUP command 169
ALTER MEMBER ADD COLS command (SQL replication) 11
ALTER MONITOR SUSPENSION command 169
ALTER MONITOR SUSPENSION TEMPLATE command 171
ALTER PUB command (event publishing) 140
ALTER PUBQMAP command (event publishing) 139
ALTER QSUB (bidirectional replication) 109
ALTER QSUB (peer-to-peer replication) 111
ALTER QSUB (unidirectional Q replication) 85
ALTER REGISTRATION command (SQL replication) 12
ALTER SUBSCRIPTION SET command (SQL replication) 15
ALTER XML PUB command (event publishing) 140
ASNCLP
PROMOTE PUB command 151
PROMOTE PUBQMAP 152
SET PROFILE (SQL replication) 48
ASNCLP commands
ALTER ALERT CONDITIONS FOR APPLY 159
ALTER ALERT CONDITIONS FOR CAPTURE 161
ALTER ALERT CONDITIONS FOR QAPPLY 164
ALTER ALERT CONDITIONS FOR QCAPTURE 166
ALTER CAPTURE PARAMETERS (Classic replication) 84
ALTER CONFIGURATION APPLY 85
ALTER CONTACT 168
ALTER GROUP 169
ALTER MEMBER ADD COLS (SQL replication) 11
ALTER MONITOR SUSPENSION 169
ASNCLP commands (continued)
ALTER MONITOR SUSPENSION TEMPLATE 171
ALTER PUB command (event publishing) 140
ALTER PUBQMAP (event publishing) 139
ALTER QSUB (bidirectional replication) 109
ALTER QSUB (peer-to-peer replication) 111
ALTER QSUB (unidirectional Q replication) 85
ALTER REGISTRATION (SQL replication) 12
ALTER SUBSCRIPTION SET (SQL replication) 15
ALTER XML PUB (event publishing) 140
ASNCLP SESSION SET TO (SQL replication) 17
CREATE ALERT CONDITIONS FOR APPLY 172
CREATE ALERT CONDITIONS FOR CAPTURE 174
CREATE ALERT CONDITIONS FOR QAPPLY 176
CREATE ALERT CONDITIONS FOR QCAPTURE 177
CREATE CONTACT 179
CREATE CONTROL TABLES (SQL replication) 180
CREATE CONTROL TABLES FOR (SQL replication) 18
CREATE GROUP 182
CREATE MEMBER (SQL replication) 20
CREATE MONITOR SUSPENSION 182
CREATE MONITOR SUSPENSION TEMPLATE 184
CREATE PUB command (event publishing) 144
CREATE PUBQMAP (event publishing) 142
CREATE QSUB (bidirectional replication) 113
CREATE QSUB (peer-to-peer replication) 116
CREATE QSUB (unidirectional Q replication) 89
CREATE REGISTRATION (SQL replication) 28
CREATE SUBSCRIPTION SET (SQL replication) 34
DELEGATE CONTACT 185
DROP ALERT CONDITIONS FOR APPLY 185
DROP ALERT CONDITIONS FOR CAPTURE 186
ASNCLP commands (continued)
DROP ALERT CONDITIONS FOR QAPPLY 186
DROP ALERT CONDITIONS FOR QCAPTURE 186
DROP CONTACT 187
DROP CONTROL TABLES (SQL replication) 36
DROP GROUP 187
DROP MEMBER (SQL replication) 37
DROP MONITOR SUSPENSION 188
DROP MONITOR SUSPENSION TEMPLATE 188
DROP PUB command (event publishing) 148
DROP PUBQMAP (event publishing) 148
DROP QSUB (unidirectional Q replication) 102
DROP REGISTRATION (SQL replication) 38
DROP STMT (SQL replication) 38
DROP SUBGROUP (multidirectional Q replication) 118
DROP SUBSCRIPTION SET (SQL replication) 39
DROP SUBTYPE (bidirectional replication) 118
DROP SUBTYPE (peer-to-peer replication) 119
DROP XML PUB (event publishing) 148
DROP interactive mode, using 2
LIST MONITOR SUSPENSION 189
LIST MONITOR SUSPENSION TEMPLATE 189
LIST PUBQMAPS 150
LIST PUBS 149
LIST QSUB (Q replication) 103
LIST REPLQMAP (Q replication) 105
LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication) 120
multidirectional Q replication 107, 132
OFFLINE LOAD (SQL replication) 40
PROMOTE REGISTRATION (SQL replication) 41
PROMOTE SUBSCRIPTION SET (SQL replication) 43
Q replication and Classic replication overview 57
Replication Alert Monitor 155
SET BIDI NODE 121
SET BIDIRECTIONAL NODE 121
SET CAPTURE SCHEMA (SQL replication) 44
SET CONNECTION (multidirectional Q replication) 123
SET DROP (SQL replication) 45
ASNCLP commands (continued)
SET DROP (unidirectional replication) 106
SET ENFORCE MATCHING CONSTRAINTS (multidirectional Q replication) 124
SET LOG (SQL replication) 46
SET MULTIDIR SCHEMA (multidirectional Q replication) 124
SET OUTPUT (monitor) 189
SET OUTPUT (SQL replication) 47
SET OUTPUT command (multidirectional Q replication) 125
SET PEER NODE 125
SET PEER TO PEER NODE 125
SET REFERENCE TABLE command (multidirectional Q replication) 127
SET RUN SCRIPT (SQL replication) 51
SET SERVER (multidirectional Q replication) 128
SET SERVER (Replication Alert Monitor) 190
SET SERVER (SQL replication) 54
SET SUBGROUP (multidirectional Q replication) 129
SET TABLES (multidirectional Q replication) 130
SET TRACE (SQL replication) 56
SQL replication 7
START PUB (event publishing) 144
START PUB command (event publishing) 153
START XML PUB (event publishing) 153
STOP PUB command (event publishing) 153
STOP XML PUB (event publishing) 153
SUBSTITUTE CONTACT 191
unidirectional Q replication 82, 131
using input file 3
ASNCLP program
getting started 1
ASNCLP scripts, using 3
ASNCLP SESSION SET TO command (SQL replication) 17

CREATE CONTACT command 179
CREATE CONTROL TABLES command (SQL replication) 180
CREATE CONTROL TABLES FOR command (SQL replication) 18
CREATE GROUP command 182
CREATE MEMBER command (SQL replication) 20
CREATE MONITOR SUSPENSION command 182
CREATE MONITOR SUSPENSION TEMPLATE command 184
CREATE PUB command (event publishing) 144
CREATE PUBQMAP command (event publishing) 142
CREATE QSUB command (bidirectional replication) 113
CREATE QSUB command (peer-to-peer replication) 116
CREATE QSUB command (unidirectional Q replication) 89
CREATE REGISTRATION command (SQL replication) 28
CREATE STMT command (SQL replication)
ASNCLP commands 32
CREATE STMT (SQL replication) 32
CREATE SUBSCRIPTION SET command (SQL replication) 34

DELEGATE CONTACT command 185
documentation accessible 193, 195
DROP ALERT CONDITIONS FOR APPLY command 185
DROP ALERT CONDITIONS FOR CAPTURE command 186
DROP ALERT CONDITIONS FOR QCAPTURE command 186
DROP ALERT CONDITIONS FOR QAPPLY command 186
DROP CONTACT command 187
DROP CONTROL TABLES command (SQL replication) 36
DROP GROUP command 187
DROP MEMBER command (SQL replication) 37
DROP MONITOR SUSPENSION command 188
DROP MONITOR SUSPENSION TEMPLATE command 188
DROP PUB command (event publishing) 148
DROP PUBQMAP command (event publishing) 148
DROP QSUB command (unidirectional Q replication) 102
DROP REGISTRATION command (SQL replication) 38
DROP STMT command (SQL replication) 38
DROP SUBGROUP command (multidirectional Q replication) 118
DROP SUBSCRIPTION SET command (SQL replication) 39
DROP SUBTYPE command (bidirectional replication) 115
DROP SUBTYPE command (peer-to-peer replication) 119
DROP XML PUB command (event publishing) 148

E
event publishing
sample scripts 135, 136

I
input file for commands
ASNCLP commands 3
interactive mode 2
introduction, ASNCLP program 1

J
Java environment, setting up 1

L
legal notices 197
LIST MONITOR SUSPENSION command 189
LIST MONITOR SUSPENSION TEMPLATE command 189
LIST PUBQMAPS command 150
LIST PUBS command 149
LIST QSUB command (Q replication) 103
LIST REPLQMAP command (Q replication) 105
LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication) 120

M
multidirectional Q replication commands 107, 132

O
OFFLINE LOAD command (SQL replication) 40
operating systems, supported 1

P
peer-to-peer configurations
promoting 80
peer-to-peer Q replication (three servers)
sample scripts 73
peer-to-peer Q replication (two servers)
sample scripts 69
PROMOTE PUB command 151
PROMOTE PUBQMAP command 152

B
batch mode, using 3
bidirectional Q replication
sample scripts 64
binding 2

C
CLASSPATH environment variable 1
CREATE ALERT CONDITIONS FOR APPLY command 172
CREATE ALERT CONDITIONS FOR CAPTURE command 174
CREATE ALERT CONDITIONS FOR QAPPLY command 176
CREATE ALERT CONDITIONS FOR QCAPTURE command 177

D
PROMOTE REGISTRATION command (SQL replication) 41
PROMOTE SUBSCRIPTION SET command (SQL replication) 43
Promoting publications 151
Promoting publishing queue maps 152
Promoting peer-to-peer configurations sample scripts 80
Promoting unidirectional configurations 79

Q
Q subscriptions
ALTER QSUB 85
changing 85

R
Replication Alert Monitor
ASNCLP commands 155
sample scripts 156

S
sample scripts
bidirectional Q replication 64
event publishing 135, 136
peer-to-peer Q replication (three servers) 73
peer-to-peer Q replication (two servers) 69
Promoting peer-to-peer configurations 80
Replication Alert Monitor 156
SQL replication 8
unidirectional Q replication 57, 61
screen readers 193, 195
SET BIDI NODE command 121
SET BIDIRECTIONAL NODE command 121
SET CAPTURE SCHEMA command (SQL replication) 44
SET CONNECTION command (multidirectional Q replication) 123
SET DROP command (SQL replication) 45
SET DROP command (unidirectional replication) 106
SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication) 124
SET LOG command (SQL replication) 46
SET MULTIDIR SCHEMA command (multidirectional Q replication) 124
SET OUTPUT command (monitor) 189
SET OUTPUT command (multidirectional Q replication) 125
SET OUTPUT command (SQL replication) 47
SET PEER NODE command 125
SET PEER_TO_PEER NODE command 125

SET PROFILE command (SQL replication) 48
SET REFERENCE TABLE command (multidirectional Q replication) 127
SET RUN SCRIPT command (SQL replication) 51
SET SERVER command (multidirectional Q replication) 128
SET SERVER command (Replication Alert Monitor) 190
SET SERVER command (SQL replication) 54
SET SUBGROUP command (multidirectional Q replication) 129
SET TABLES command (multidirectional Q replication) 130
SET TRACE command (SQL replication) 56
SQL replication
ASNCLP commands 7
sample scripts 8
START PUB command (event publishing) 153
START XML PUB command (event publishing) 153
STOP PUB command (event publishing) 153
STOP XML PUB command (event publishing) 153
SUBSTITUTE CONTACT command 191
supported operating systems 1

T
trademarks 199

U
unidirectional Q replication commands 82, 131
sample scripts 57, 61

Z
z/OS packages, binding 2