



ASNCLP Program Reference for Replication and Event Publishing



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 291.

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	3
Running the ASNCLP commands in execute-immediately mode	3
Running the ASNCLP commands by using an input file	4
ASNCLP configuration file	4

Chapter 2. ASNCLP commands for SQL replication.

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	33
CREATE SUBSCRIPTION SET command	35
DROP CONTROL TABLES ON command	37
DROP MEMBER command	38
DROP REGISTRATION command	39
DROP STMT command	39
DROP SUBSCRIPTION SET command	40
OFFLINE LOAD command	41
PROMOTE REGISTRATION command	41
PROMOTE SUBSCRIPTION SET command	43
SET CAPTURE SCHEMA command (SQL replication)	45
SET DROP command (SQL replication)	46
SET LOG command	47
SET OUTPUT command (SQL replication)	48
SET PROFILE command (SQL replication)	48
SET RUN SCRIPT command (SQL replication)	52
SET SERVER command (SQL replication)	55
SET TRACE command	58

Chapter 3. Sample ASNCLP scripts for Q replication

Sample ASNCLP scripts for setting up unidirectional Q replication	59
Sample ASNCLP scripts for setting up unidirectional Q replication from a Classic data source	62

Sample ASNCLP scripts for setting up bidirectional Q replication	65
Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers)	70
Sample ASNCLP scripts for setting up peer-to-peer Q replication (three servers)	75
Sample ASNCLP script for promoting unidirectional configurations	81
Sample ASNCLP scripts for promoting peer-to-peer configurations	82

Chapter 4. ASNCLP commands for unidirectional Q replication

ALTER ADD COLUMN command	86
ALTER CAPTURE PARAMETERS command (Classic replication)	88
ALTER CONFIGURATION APPLY command	88
ALTER REPLQMAP command	89
ALTER QSUB command (unidirectional replication)	91
ASNCLP SESSION SET TO command	95
CREATE CONTROL TABLES FOR command	96
CREATE REPLQMAP command	102
CREATE QSUB command (unidirectional replication)	103
DROP CONTROL TABLES ON command	119
DROP REPLQMAP command	119
DROP QSUB command (unidirectional Q replication)	120
LIST QSUB command (Q replication)	121
LIST APPLY SCHEMA command	122
LIST CAPTURE SCHEMA command	123
LIST REPLQMAP command (Q replication)	124
LOAD DONE command	126
PROMOTE QSUB command	127
PROMOTE REPLQMAP command	129
SET APPLY SCHEMA command	130
SET CAPTURE SCHEMA command	130
SET DROP command (unidirectional replication)	131
SET LOG command	132
SET OUTPUT command	133
SET PROFILE command	133
SET QMANAGER command	137
SET RUN SCRIPT command	138
SET SERVER command	140
SET TRACE command	144
SHOW SET ENV command	144
START QSUB command	144
STOP QSUB command	146
VALIDATE WSMQ ENVIRONMENT FOR command	147
VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command	148

Chapter 5. ASNCLP commands for multidirectional Q replication

ALTER REPLQMAP command	152
ALTER QSUB command (bidirectional replication)	154
ALTER QSUB command (peer-to-peer replication)	157
ASNCLP SESSION SET TO command	158
CREATE CONTROL TABLES FOR command	159
CREATE REPLQMAP command	165
CREATE QSUB command (bidirectional replication)	167
CREATE QSUB command (peer-to-peer replication)	171
DROP CONTROL TABLES ON command	174
DROP REPLQMAP command	175
DROP SUBGROUP command (multidirectional Q replication)	175
DROP SUBTYPE command (bidirectional replication)	176
DROP SUBTYPE command (peer-to-peer replication)	176
LIST APPLY SCHEMA command	177
LIST CAPTURE SCHEMA command	178
LOAD DONE command	179
LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication)	180
PROMOTE QSUB command	182
PROMOTE REPLQMAP command	183
SET APPLY SCHEMA command	185
SET BIDI NODE command	185
SET CAPTURE SCHEMA command	187
SET CONNECTION command (multidirectional Q replication)	188
SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication)	189
SET LOG command	189
SET MULTIDIR SCHEMA command (multidirectional Q replication)	190
SET OUTPUT command (multidirectional Q replication)	190
SET PEER NODE command	191
SET PROFILE command	193
SET QMANAGER command	196
SET REFERENCE TABLE command (multidirectional Q replication)	197
SET SERVER command (multidirectional Q replication)	198
SET SUBGROUP command (multidirectional Q replication)	200
SET TABLES command (multidirectional Q replication)	200
SET TRACE command	201
SHOW SET ENV command	202
START QSUB command	202
STOP QSUB command	203
VALIDATE WSMQ ENVIRONMENT FOR command	205
VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command	206

Chapter 6. ASNCLP commands for event publishing 207

Sample ASNCLP scripts for setting up event publishing	208
ALTER ADD COLUMN command	211

ALTER PUB command	212
ALTER PUBQMAP command	214
ASNCLP SESSION SET TO command	215
CREATE CONTROL TABLES FOR command	216
CREATE PUB command	222
CREATE PUBQMAP command	226
DROP CONTROL TABLES ON command	228
DROP PUB command	228
DROP PUBQMAP command	229
LIST CAPTURE SCHEMA command	229
LIST PUBS command	230
LIST PUBQMAPS command	231
PROMOTE PUB command	232
PROMOTE PUBQMAP command	234
SET CAPTURE SCHEMA command	235
SET LOG command	236
SET OUTPUT command	236
SET QMANAGER command	237
SET RUN SCRIPT command	238
SET SERVER command (event publishing)	240
SET TRACE command	243
START PUB command	244
STOP PUB command	244
VALIDATE WSMQ ENVIRONMENT FOR command	244

Chapter 7. ASNCLP commands for the Replication Alert Monitor 247

Sample ASNCLP scripts for setting up the Replication Alert Monitor	248
ALTER ALERT CONDITIONS FOR APPLY command	251
ALTER ALERT CONDITIONS FOR CAPTURE command	253
ALTER ALERT CONDITIONS FOR QAPPLY command	256
ALTER ALERT CONDITIONS FOR QCAPTURE command	258
ALTER CONTACT command	260
ALTER GROUP command	261
ALTER MONITOR SUSPENSION command	261
ALTER MONITOR SUSPENSION TEMPLATE command	262
CREATE ALERT CONDITIONS FOR APPLY command	263
CREATE ALERT CONDITIONS FOR CAPTURE command	266
CREATE ALERT CONDITIONS FOR QAPPLY command	267
CREATE ALERT CONDITIONS FOR QCAPTURE command	269
CREATE CONTACT command	271
CREATE CONTROL TABLES FOR command	272
CREATE GROUP command	274
CREATE MONITOR SUSPENSION command	274
CREATE MONITOR SUSPENSION TEMPLATE command	276
DELEGATE CONTACT command	277
DROP ALERT CONDITIONS FOR APPLY command	277

DROP ALERT CONDITIONS FOR CAPTURE command	278
DROP ALERT CONDITIONS FOR QAPPLY command	278
DROP ALERT CONDITIONS FOR QCAPTURE command	278
DROP CONTACT command	279
DROP GROUP command	279
DROP MONITOR SUSPENSION command	280
DROP MONITOR SUSPENSION TEMPLATE command	280
LIST MONITOR SUSPENSION command	281
LIST MONITOR SUSPENSION TEMPLATE command	281
SET OUTPUT command.	281
SET SERVER command	282

SUBSTITUTE CONTACT command	283
--------------------------------------	-----

Product documentation 285

Contacting IBM	285
--------------------------	-----

How to read syntax diagrams 287

Product accessibility 289

Notices 291

Trademarks	293
----------------------	-----

Index 295

Chapter 1. Getting started with the ASNCLP program

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.

Some commands are available only for specific types of replication or publishing configurations.

Supported operating systems

The ASNCLP program runs on Linux[®], UNIX[®], Windows[®], and UNIX System Services (USS) on z/OS[®]. 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 database connection statement to each of the servers.

Note: Additional configuration steps are required to enable the ASNCLP to run on USS. For details, see *Optional: Enabling the ASNCLP program to run on USS* in the *Information Management Software for z/OS Solutions Information Center*.

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.

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

Procedure

Add the following path to your PATH environment variable:

INSTDIR\java\jdk

Where *INSTDIR* is the DB2 instance directory. On Linux and UNIX, the instance directory is the *INSTDIR*/sqllib directory, where *INSTDIR* is the home directory of the instance owner. On Windows, the instance directory is the \sqllib directory where DB2 was installed.

For all DB2 products except the IBM Data Server Runtime Client, the DB2 Database for Linux, UNIX, and Windows installation process automatically installs the SDK for Java. If you need to install the SDK, go to the "IBM developer kits" page on IBM developerWorks: <http://www.ibm.com/developerworks/java/jdk/index.html>

Examples

Linux UNIX

To set the PATH environment variable from a UNIX command prompt:

```
export
CLASSPATH=$PATH
:/u/myinst/sqllib/java/jdk;
```

Windows

To set the PATH environment variable from a Windows command prompt:

```
set CLASSPATH=%PATH%;%c:\Program File\sqllib\java\jdk;
```

Binding z/OS packages for the ASNCLP program

z/OS

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:

```
bind @ddcmvs.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:

```
ASNCLP
```

The ASNCLP command starts the ASNCLP program and changes the command prompt to `Rep1 >`.

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 by 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 by 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.

Tip: You can specify that the ASNCLP program ignores some errors that it encounters when creating objects that already exist by using the `SET RUN SCRIPT LATER GENERATE SQL FOR EXISTING YES` command.

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

To access Classic sources or run on UNIX System Services (USS) for z/OS, the ASNCLP program requires connectivity information to be provided through a configuration file.

The ASNCLP configuration file contains a group of lines for each data 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 program can connect to the data 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:

Classic replication

Specify `Type=classic`.

ASNCLP on USS

If the server is DB2 for z/OS or DB2 for Linux, UNIX, and Windows and you are running the ASNCLP on USS, specify `Type=DB2`.

Data source

Specifies the location of the source data:

Classic replication

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

ASNCLP on USS

If you are running the ASNCLP on USS, for DB2 sources this parameter specifies the DB2 for z/OS location name or DB2 for Linux, UNIX, and Windows database name.

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 server where the data source resides.

Codepage

Codepage is an optional parameter for Classic sources that describes the code page of the data.

Example 1

The following example shows a configuration file that is used on USS to specify a connection to a DB2 for z/OS subsystem:

```
[DB2ZOS]
Type=DB2
Data source=dsn7
Host=stplex4a.svl.ibm.com
Port=2080
```

Example 2

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

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

```
Type=classic  
Data source=CACSALES  
Host=145.145.231.87  
Port=8095
```

Usage notes

You can save the configuration file to any location. The default 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/sqllib/classic_files/` directory.

```
SET SERVER capture TO CONFIG SERVER cacsamp1 FILE  
"/home/db2inst/sqllib/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 1. ASNCLP commands for SQL replication

If you want to ...	Use this command
Add columns to an existing member	“ALTER MEMBER ADD COLS command” on page 11
Change the properties of a registration	“ALTER REGISTRATION command” on page 12
Change the properties of a subscription set	“ALTER SUBSCRIPTION SET command” on page 15
Establish a session for SQL replication	“ASNCLP SESSION SET TO command (SQL replication)” on page 17
Create control tables	“CREATE CONTROL TABLES FOR command (SQL replication)” on page 18
Create a subscription-set member	“CREATE MEMBER command” on page 20
Create a registration	“CREATE REGISTRATION command” on page 28
Create a SQL statement that is processed with an existing subscription set	“CREATE STMT command” on page 33
Create a subscription set	“CREATE SUBSCRIPTION SET command” on page 35
Drop control tables	“DROP CONTROL TABLES ON command” on page 37
Delete a subscription-set member	“DROP MEMBER command” on page 38
Delete a registration	“DROP REGISTRATION command” on page 39
Delete SQL statements for an existing subscription set	“DROP STMT command” on page 39
Delete a subscription set	“DROP SUBSCRIPTION SET command” on page 40
Control a manual full refresh for offline load procedures	“OFFLINE LOAD command” on page 41
Promote a registration	“PROMOTE REGISTRATION command” on page 41
Promote a subscription set	“PROMOTE SUBSCRIPTION SET command” on page 43
Set a source and target Capture schema for all task commands	“SET CAPTURE SCHEMA command (SQL replication)” on page 45
Specify whether to drop the table space when you drop the replication object that it contains	“SET DROP command (SQL replication)” on page 46
Set the log file name for the ASNCLP program	“SET LOG command” on page 47
Specify a name for the output files that contain the SQL scripts	“SET OUTPUT command (SQL replication)” on page 48
Set up customization rules for creating table space objects	“SET PROFILE command (SQL replication)” on page 48
Specify whether to automatically run the SQL statements before the ASNCLP commands process the next task command	“SET RUN SCRIPT command (SQL replication)” on page 52

Table 1. ASNCLP commands for SQL replication (continued)

If you want to ...	Use this command
Specify the server (database) used in the ASNCLP session, authentication information, and other required parameters for connecting to the server	“SET SERVER command (SQL replication)” on page 55
Enable and disable the tracing for the ASNCLP commands	“SET TRACE command” on page 58

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 one SQL script that creates definitions at the Capture control server and the Apply control server).
5. Subscription-set member (generates three SQL scripts that create definitions at the Capture control server and the Apply control server and create a target table and table space at 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

```
# 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.

CREATE CONTROL TABLES FOR CAPTURE SERVER;

# 3 Ending the ASNCLP session.

QUIT;

```

ASNCLP script 2 (registration)

This script generates SQL statements that register the PROJECT 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 PROJECT 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 CDPROJECT. Because the COLS keyword is not specified, all columns
# in the source table are registered.

CREATE REGISTRATION (DB2ADMIN.Project) DIFFERENTIAL REFRESH STAGE CDPROJECT;

# 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;

```

```
# 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 one script, ct1subset.sql, which inserts
# the subscription set definition in the IBMSNAP_SUBS_SET table.

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 CONTROL SCRIPT "ct1subset.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 A000. 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 A000 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 for the registered source table PROJECT. 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 three SQL scripts, capmember.sql and
# appmember.sql to define the subscription set member in the Capture and
# Apply control tables, and projtrg.sql to create a target table and table
# space.

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"
TARGET SCRIPT "projtrg.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, A000 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 "c:\TSTRG.TS" SIZE 700 PAGES;
CREATE MEMBER IN SETNAME SET00 APPLYQUAL A000 ACTIVATE YES SOURCE PROJECT
TARGET NAME TRGPROJECT DEFINITION IN TSTRG00 CREATE USING PROFILE TBSPROFILE
TYPE USERCOPY COLS ALL REGISTERED;
```

```
# 3 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

Output file	Contains SQL to ...
capctrl.sql	Create Capture control tables
register.sql	Register a source table
appctrl.sql	Create Apply control tables
ctlssubset.sql	Insert definitions for a subscription set into the Apply control tables
capmember.sql	Insert definitions for a subscription-set member into the Capture control tables
appmember.sql	Insert definitions for a subscription-set member into the Apply control tables
projtrg.sql	Create a target table and table space

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—▶▶
▶—objname—TARGET—objname—▶
  └─objowner—┘      └─objowner—┘
▶—COLS—(—EXPRESSION—"source-col-or-expr"—TARGET—name—+)—▶▶
```

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-expr"

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 A000 SOURCE DB2ADMIN.STAFF  
TARGET DB2ADMIN.TRGSTAFF COLS (EXPRESSION "source-col-or-expr" 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

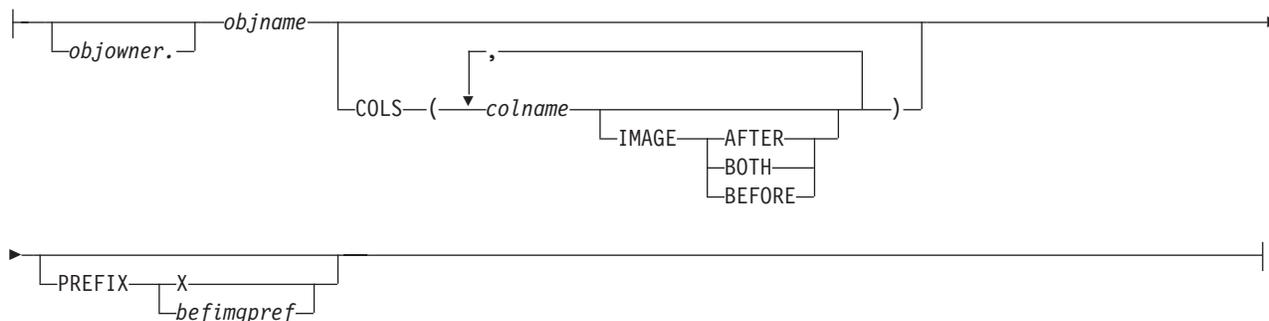
```
▶▶ ALTER REGISTRATION { ROW | ADD } row-clause | add-cols-clause
```

row-clause:

```
( ( objowner . objname ) [ CONFLICT { NONE | STANDARD | ENHANCED } ]  
[ UPDATE AS DELETE INSERT { OFF | ON } ] [ CAPTURE { ALL | CHANGES } ] [ FORWARDING { OFF | ON } ]
```



add-cols-clause:



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 by 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 by 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:

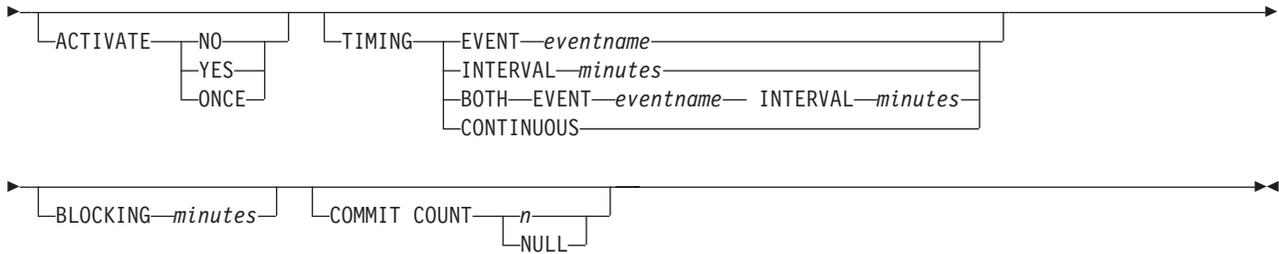
```
ALTER REGISTRATION ADD DB2ADMIN.STAFF COLS (C002 IMAGE BOTH)
```

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—  
R—  
U—  
F ONLY—  
S ONLY—  
P—
```



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 master 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_SYNC_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 within the AQ00 Apply qualifier to a read-only subscription set type 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 REPLICATION—▶▶

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

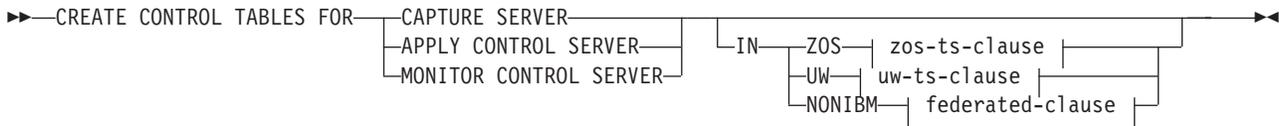
To set the ASNCLP session to SQL replication:

```
ASNCLP SESSION SET TO SQL REPLICATION
```

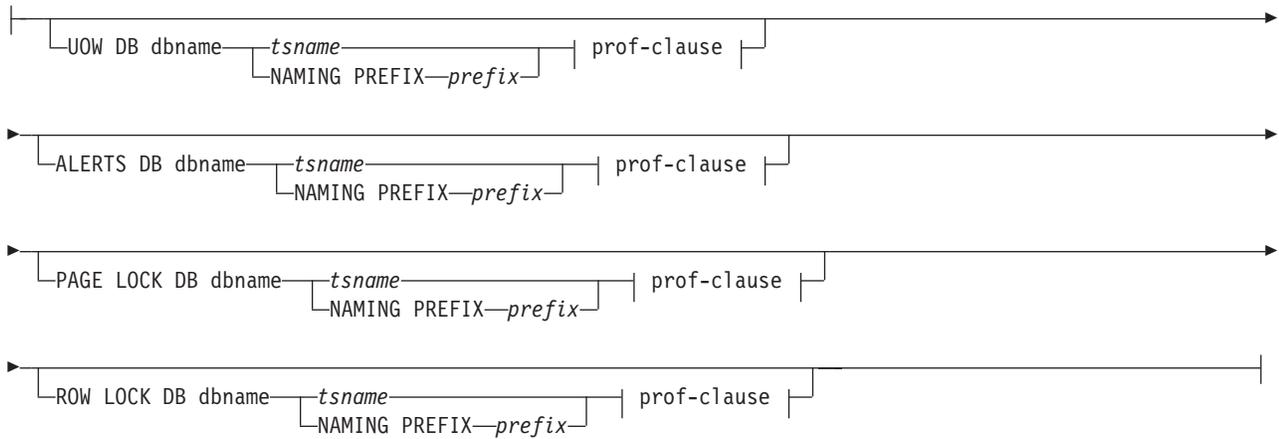
CREATE CONTROL TABLES FOR command (SQL replication)

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

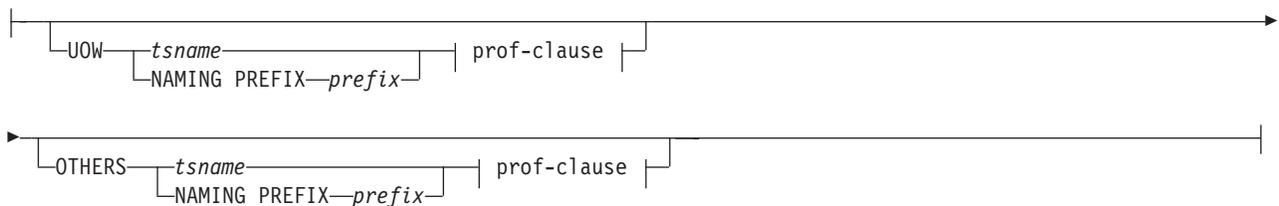
Syntax



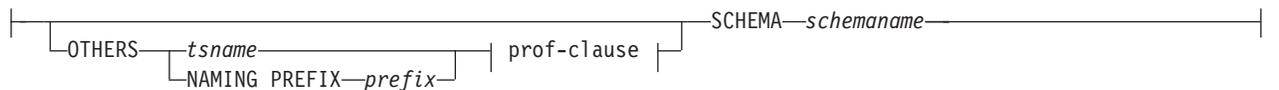
zos-ts-clause:



uw-ts-clause:



federated-clause:



prof-clause:



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 uppercase. 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*

z/OS 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 and use 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
```

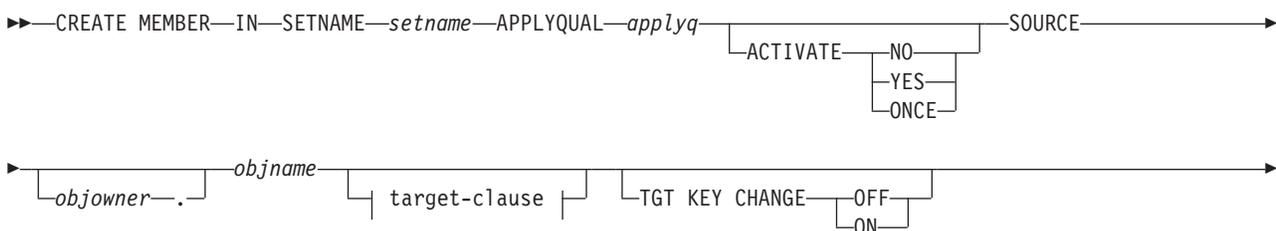
CREATE MEMBER command

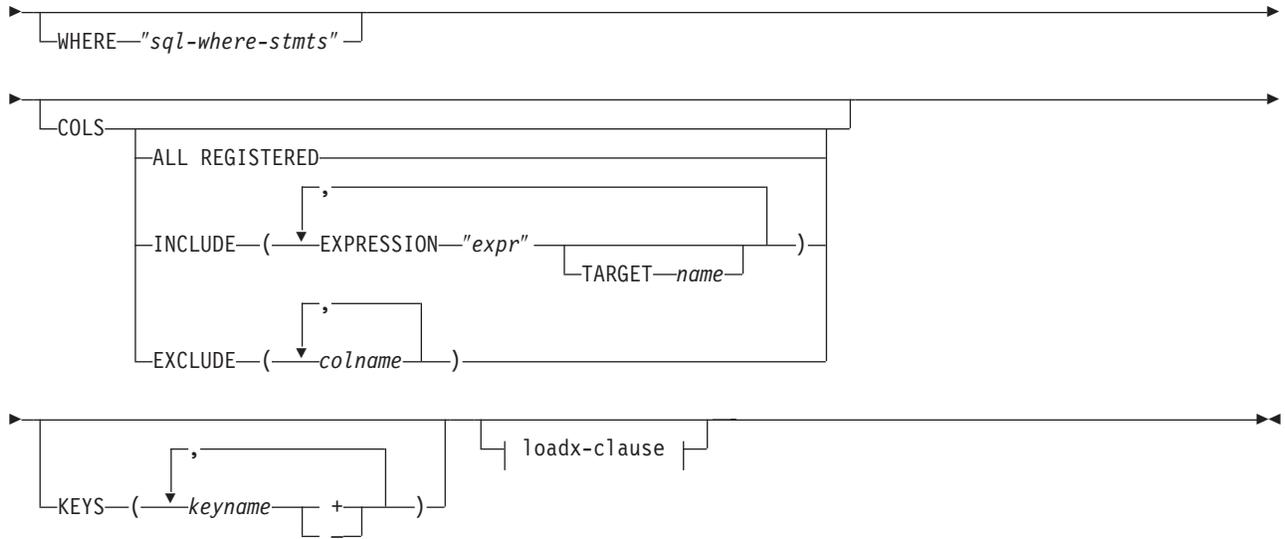
Use the CREATE MEMBER command to add a subscription-set member to an existing subscription set.

Adding a member to a set includes:

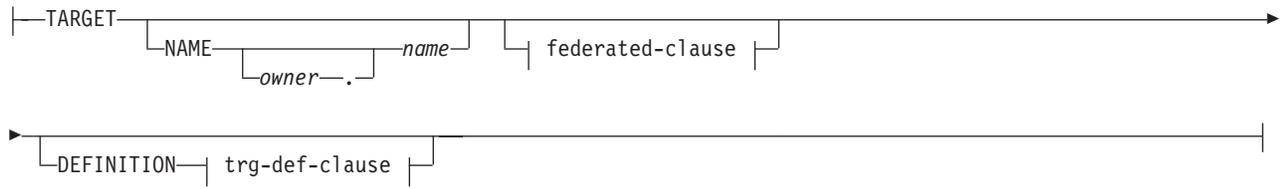
- Creating the mapping between the source and target tables (database objects).
- Creating the mapping between the source and target columns.
- Creating the target table (database object), if it doesn't already exist.
- Creating the target index, if necessary.
- Setting the KEYS value for the index.

Syntax





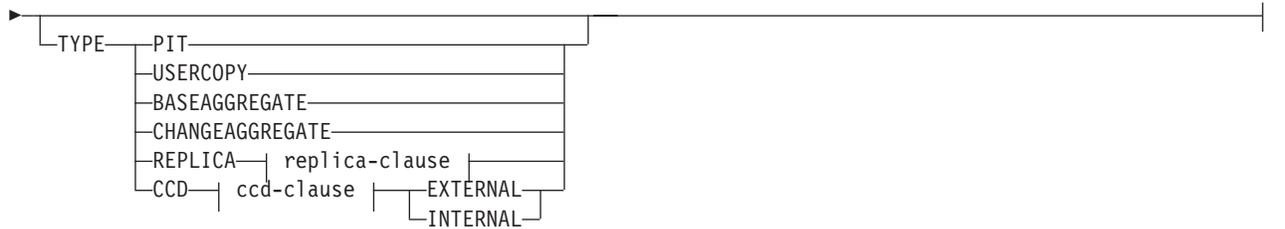
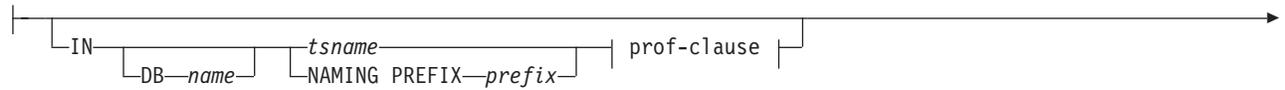
target-clause:



federated-clause:



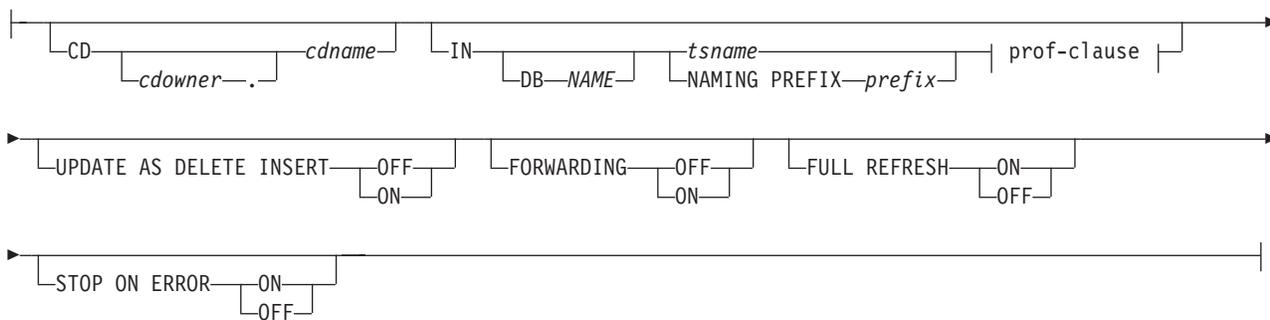
trg-def-clause:



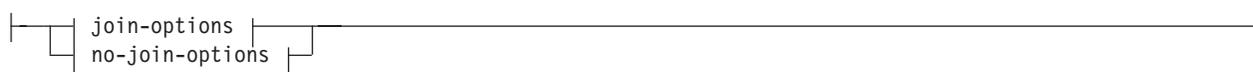
prof-clause:



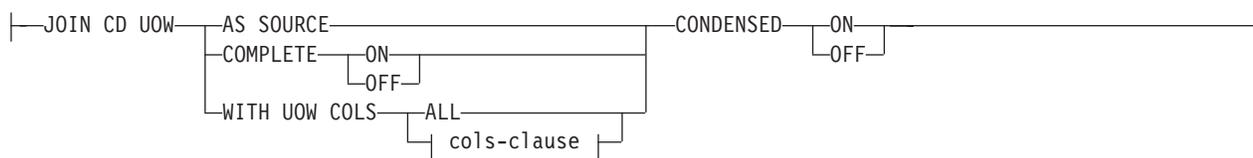
replica-clause:



ccd-clause:



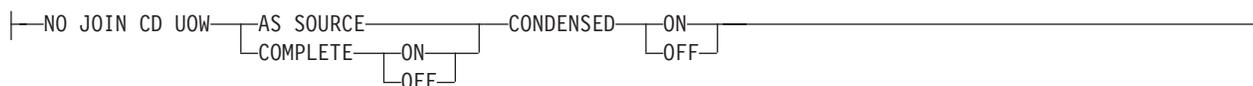
join-options:



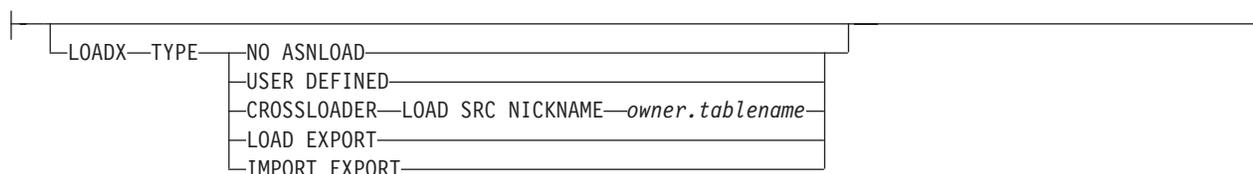
cols-clause:



no-join-options:



loadx-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 *"column_or_expression"*

The EXPRESSION keyword must precede the name of any source column that you want to include in the target table, or any SQL expression that you use to transform data between the source and target. Surround column names or expressions with double quotation marks ("). Separate multiple columns or expressions by using commas. The following example specifies that you want to include columns C1 and C2 from the source table:

```
COLS INCLUDE (EXPRESSION "C1", EXPRESSION "C2")
```

TARGET *column_name*

You must use the TARGET keyword in the following cases:

- An expression is specified in the COLS INCLUDE statement. The TARGET keyword specifies the column or columns in the target table to which you want the results of the expression applied.
- The target table already exists, a regular source column name is used in the COLS INCLUDE statement, and the target column name is different from the source column name.

The following example specifies that you want to include two columns and an expression from the source table: column C1, column C2 mapped to a column named TGTC2 at the target, and an expression

that concatenates the values in columns C3 and C4 from the source table and applies the new value into the column C3C4 at the target:
COLS INCLUDE (EXPRESSION "C1", EXPRESSION "C2" TARGET "TGTC2",
EXPRESSION "C3||C4" TARGET "C3C4")

EXCLUDE (*column_name*)

Specify to exclude one or more source columns from the target table definition. You can only use this keyword when you are creating a new target table, or for an existing target table when the source and target tables have the same column names.

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.

DEFINITION

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

federated-clause

REMOTE SCHEMA *owner*

Specifies the schema of a new target table that is created by the ASNCLP. If this keyword is not used, the default schema is the remote authorization ID for the non-DB2 target database.

REMOTE TABLE *name*

Specifies the name of a new target table that is created by the ASNCLP. If this keyword is not used, the default table name is the name of the corresponding nickname in the federated database.

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 changes.

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

Linux UNIX Windows Specify to use an EXPORT/LOAD combination for this member.

IMPORT EXPORT

Linux UNIX Windows 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 AQ00 SOURCE DB2ADMIN.STAFF
    TARGET NAME DB2ADMIN.TRGSTAFF DEFINITION IN TSUOW100 CREATE USING
    PROFILE TBSPROFILE;
```

Linux UNIX Windows

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: TRGEMPENICK

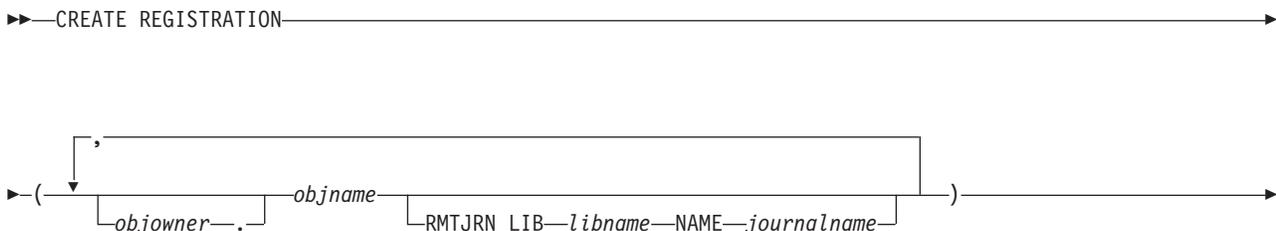
The commands create definitions for a new target table in the SQL Server database with a remote schema of dbo and a name of TRGEMPLOYEE.

```
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.TRGEMPENICK
    REMOTE SCHEMA dbo REMOTE TABLE TRGEMPLOYEE;
```

CREATE REGISTRATION command

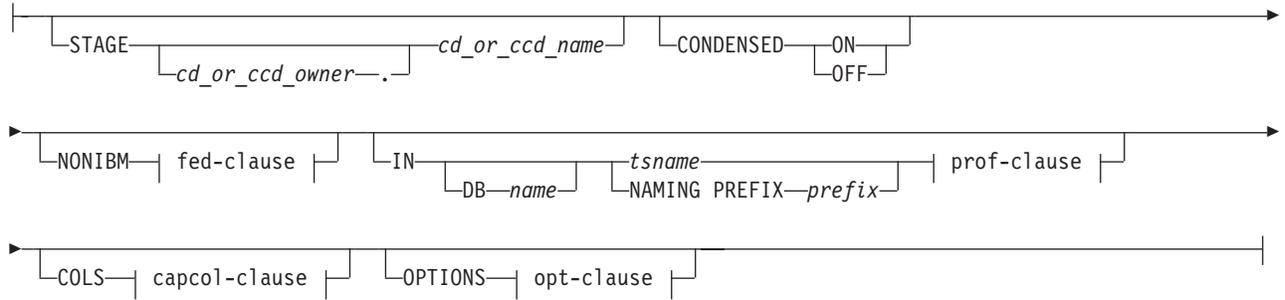
Use the CREATE REGISTRATION command to register one or more source tables, views, or nicknames for replication.

Syntax

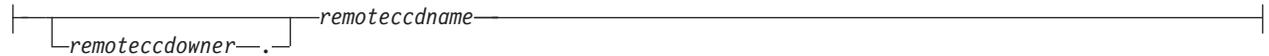




diff-ref-clause:



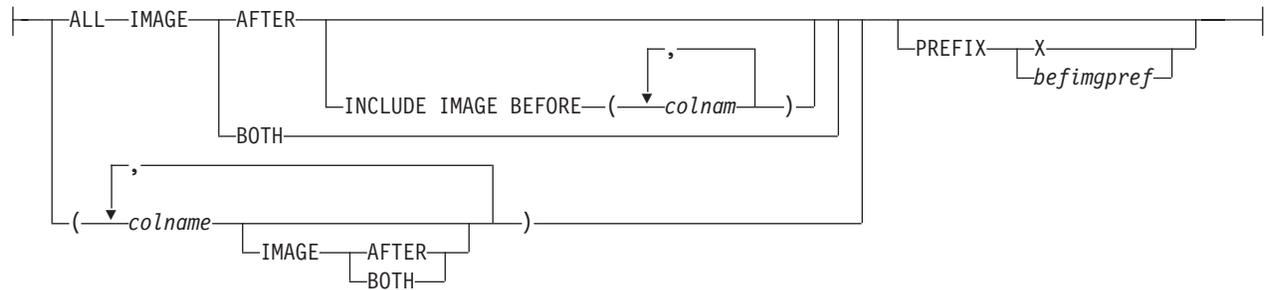
fed-clause:



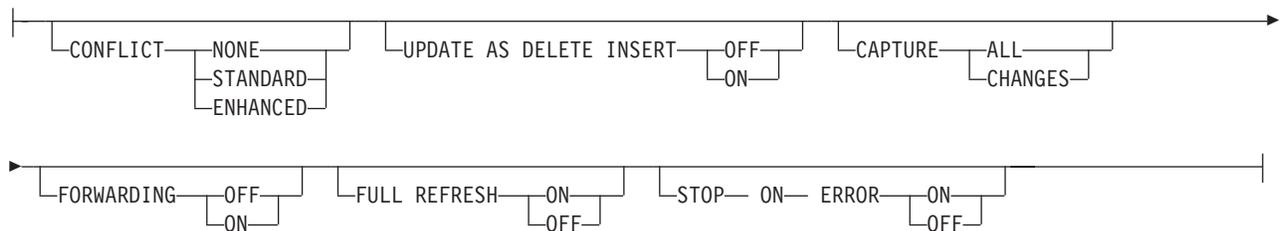
prof-clause:



capcol-clause:



opt-clause:



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*

System i

Specifies the System i library name.

NAME *journalname*

System i

Specifies the System i 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 by 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 a table. If the object is a view, you cannot register a subset of the 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.

INCLUDE IMAGE BEFORE

Specify to register before images along with after images for the listed columns.

colname

Specifies a list of the columns for which you want to register before images.

IMAGE 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 by 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 by 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
```

Example 3

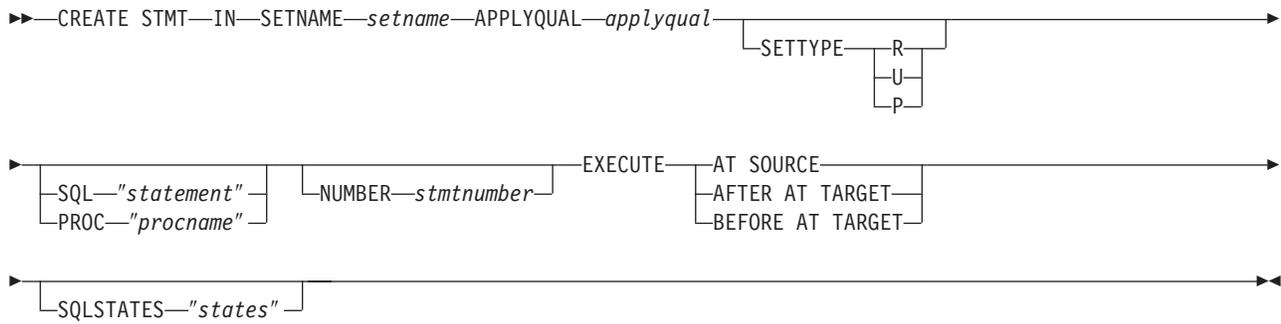
To create a registration for DB2ADMIN.EMPLOYEE that updates the target table as the source objects change, registers after-images for all of the columns in the source table, and also registers before images for the SALARY and BONUS columns:

```
CREATE REGISTRATION (DB2ADMIN.EMPLOYEE) DIFFERENTIAL REFRESH
COLS ALL IMAGE AFTER INCLUDE IMAGE BEFORE(SALARY,BONUS)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



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 "*procname*"

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 A000 SQL "statement" EXECUTE AT SOURCE
```

Example 2

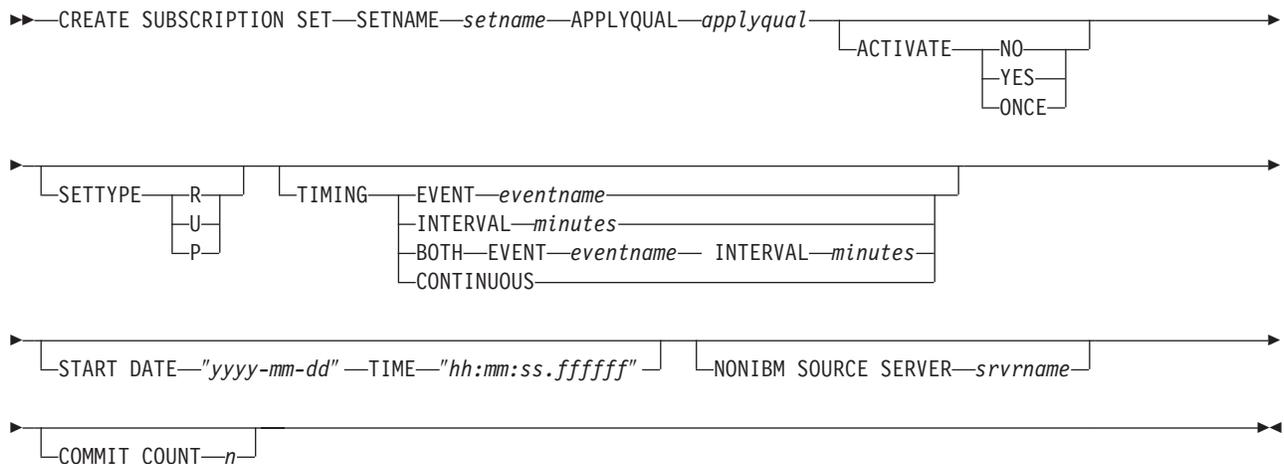
To create a statement for the SET00 subscription set that executes the stored procedure at the target server before the Apply program processes the subscription set:

```
CREATE STMT IN SETNAME SET00 APPLYQUAL AQ00 PROC "procname" EXECUTE BEFORE AT TARGET
```

CREATE SUBSCRIPTION SET command

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

Syntax



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 A000 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:

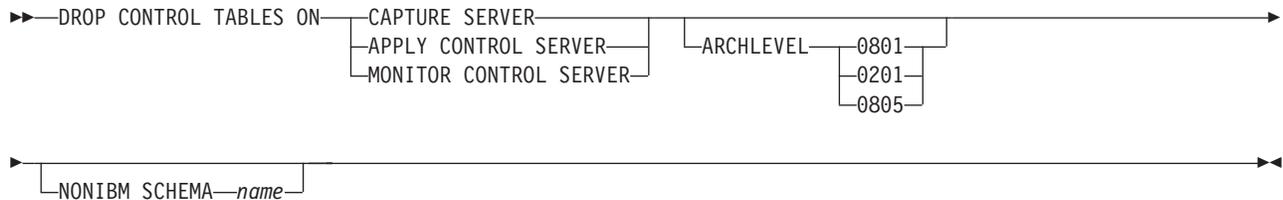
```
CREATE SUBSCRIPTION SET SETNAME SET00 APPLYQUAL AQ00 ACTIVATE ONCE TIMING CONTINUOUS
START DATE "2006-11-22" TIME "09:00:00.000000" NONIBM SOURCE SERVER SAMPLE
```

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



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 SETTYPE ( R | U | P ) ►►
```

► NUMBER ((stmtnumber))

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 A000 NUMBER (5)
```

DROP SUBSCRIPTION SET command

Use the DROP SUBSCRIPTION SET command to drop an existing subscription set for a specified Apply qualifier.

Syntax

►► DROP SUBSCRIPTION SET SETNAME *setname* APPLYQUAL *applyqual* ◀◀

Parameters

SETNAME *setname*

Specifies the subscription-set name.

APPLYQUAL *applyqual*

Specifies the Apply qualifier for the subscription set.

Usage notes

- If the subscription set has members, all members and statements will be dropped.
- See the “DROP MEMBER command” on page 38 for the rules that affect the dropped objects.

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 | AFTER } 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

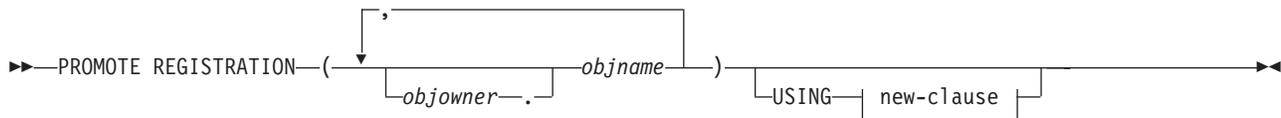
To run the OFFLINE LOAD AFTER command and to reactivate the subscription set SET00 and to reset the IBMSNAP_PRUNCNTL SET and IBMSNAP_SIGNAL tables:

```
OFFLINE LOAD AFTER SETNAME SET00 APPLYQUAL AQ00
```

PROMOTE REGISTRATION command

Use the PROMOTE REGISTRATION command to promote existing registrations.

Syntax



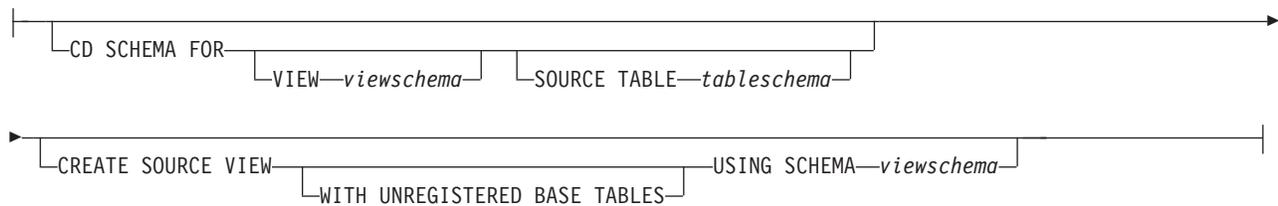
new-clause:



tbl-clause:



view-clause:



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.

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:

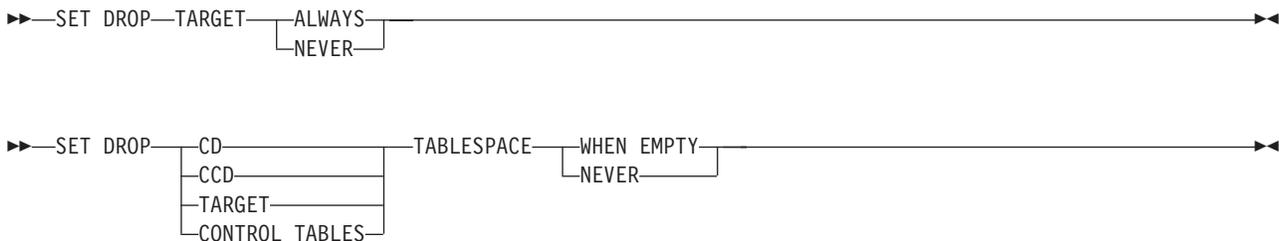
```
SET CAPTURE SCHEMA SOURCE ASN1
```

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



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

*"logfile*name"

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

```
▶▶ 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] [DB *dbname*] [BUFFERPOOL *bufferpoolname*] [ENCODING [EBCDIC | ASCII | UNICODE]]

[STOGROUP *stogroupname*] [priqty-clause] [secqty-clause]

priqty-clause:

[PRIQTY [ABSOLUTE *n* | PERCENT OF SOURCE]]

secqty-clause:

[SECQTY [ABSOLUTE *m* | PERCENT OF SOURCE]]

uw-tbs-clause:

[UW] [BUFFERPOOL *bufferpoolname*] [PAGESIZE *n*]

[USING [FILE | DEVICE] "*container*" [SIZE *n* [PAGES | 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 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*
z/OS 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-clause

PRIQTY

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

ABSOLUTE

Specifies an actual value in kilobytes (denoted as *n* in the syntax diagram) for primary 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 SYSSTAT.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.

secqty-clause

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 *m* in the syntax diagram) for secondary 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 SYSSTAT.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.

Restriction: The page size of the table space must match the page size of the buffer pool.

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 3

To undo the profile TBSPROFILE:

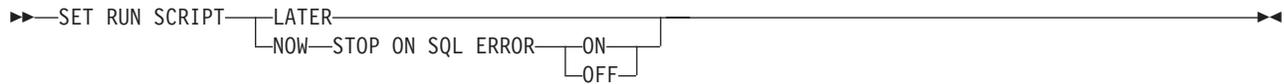
```
SET PROFILE TBSPROFILE UNDO
```

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 53 helps you understand when to run commands immediately and when to run them later.

Syntax



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 54 shows these dependencies for SQL replication.

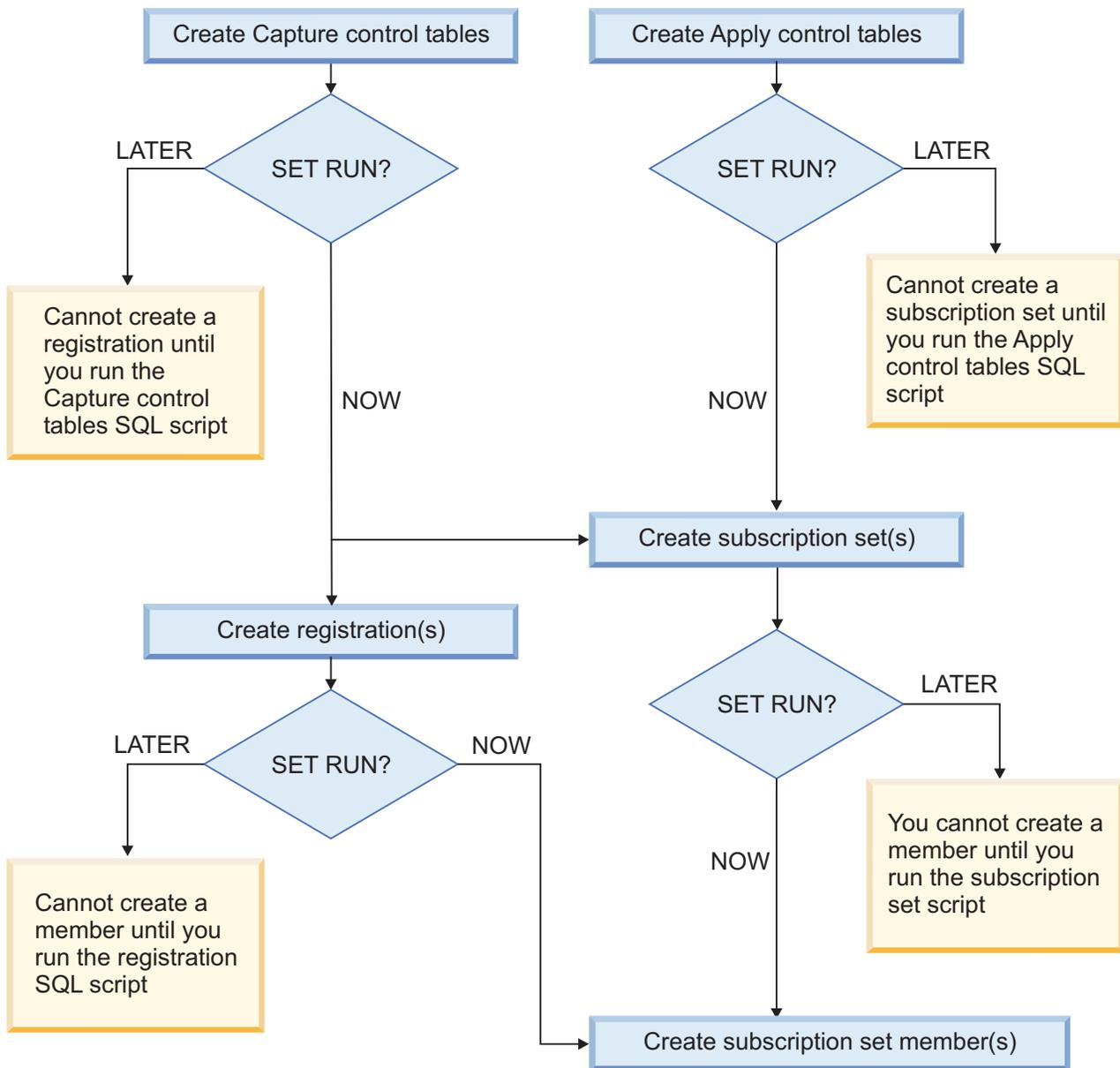


Figure 1. Dependencies between ASNCLP commands for SQL replication. This diagram shows the dependencies between ASNCLP CREATE commands that are used to set up SQL replication. It assumes all objects use the default schema of ASN.

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 System i 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.

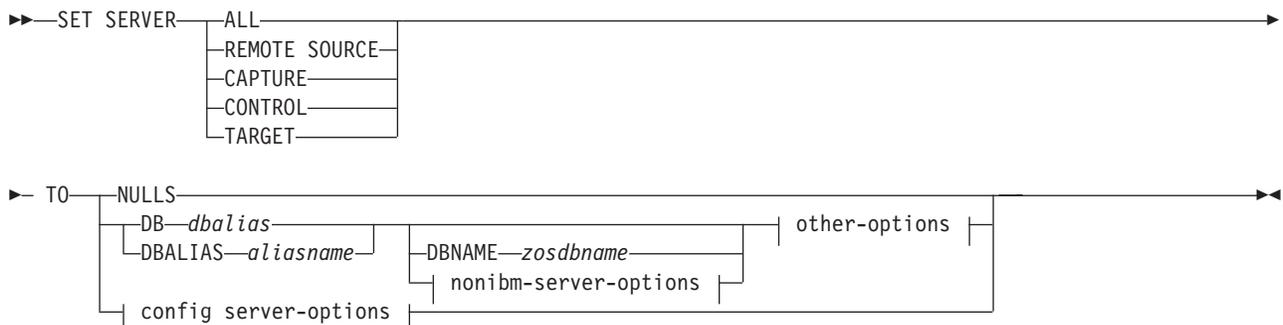
All registration commands (including promote)

Set the Capture control server before running the registration commands. For System i, you must also set the remote source server.

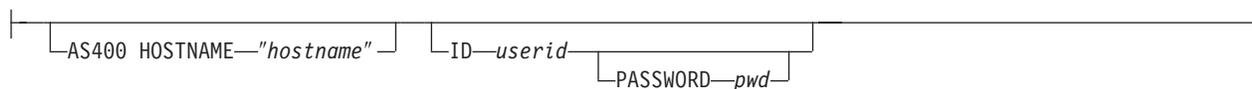
All subscription commands (including promote)

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 System i, you must set the remote source server.

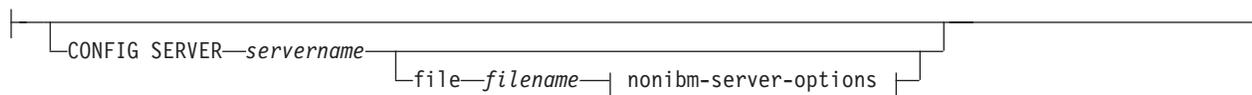
Syntax



other-options:



config server-options:



nonibm-server-options:



Parameters

ALL

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

REMOTE SOURCE

System i

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 of a z/OS subsystem or Linux, UNIX, Windows, or System i database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, Windows, or System i database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS

Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

other-options clause:

AS400 HOSTNAME "*hostname*"

System i

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.

config server-options clause:

CONFIG SERVER *servername*

DB2 sources only: Specifies the DB2 source to connect with when the ASNCLP program is running on UNIX System Services (USS) for z/OS. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the FILE parameter, the ASNCLP program attempts to use the asnservers.ini file in the current directory, if that file exists.

nonibm server-options clause

NONIBM SERVER *remsrvr*

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.

Note: If the ASNCLP is running on USS, you must specify the NONIBM SERVER keyword along with the CONFIG SERVER keyword because an input file is required to connect to the source database.

Usage notes

- Use the NONIBM SERVER clause to set up replication with 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.

Example 4

In this example, the ASNCLP program is running on USS.

Given a configuration file called `sample.ini` that contains the following information:

```
[sample1]
Type=DB2
Data source=dsn7
Host=stplex4a.svl.ibm.com
Port=2080
```

Use the following command to specify the SAMPLE database as the Capture control server:

```
SET SERVER CAPTURE TO CONFIG SERVER sample1 FILE sample.ini ID id1 PASSWORD pwd1;
```

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. 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 62 below the sample describes the SQL scripts that are generated.

For help creating the WebSphere MQ objects that are used in these scripts, see WebSphere MQ setup script generator 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:

- 1** Setting the environment
- 2** Creating Q Capture control tables
- 3** Creating Q Apply control tables
- 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 two SQL scripts: qcapctrl.sql, which creates
# Q Capture control tables at the SAMPLE database, and qappctrl.sql, which creates
# Q Apply control tables at the TARGET database.
# The SETQMANAGER commands are required for creating Q replication control tables.
```

```
# The SET RUN SCRIPT LATER option allows you to review the SQL scripts before
# they are run.
```

```
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "qcontrol.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;
```

```
# 2 Creating Q Capture control tables.
# The command specifies a restart queue and administration queue, doubles the
# default amount of memory available to build transactions to 64 MB, and reduces
# the default interval for recording performance information to 600000 milliseconds
# (one minute).
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "ASN1.QM1.RESTARTQ" ADMINQ "ASN1.QM1.ADMINQ"
MEMORY LIMIT 64 MONITOR INTERVAL 600000 IN UW TBSPACE TSQCAP;
```

```
# 3 Creating Q Apply control tables.
# This command specifies a password file, asnpwd.aut. The Q Apply program uses this
# file to connect to the Q Capture server when it loads the target table.
```

```
CREATE CONTROL TABLES FOR APPLY SERVER USING PWDFILE "asnpwd.aut"
IN UW TBSPACE TSQAPP;
```

```
# 4 Ending the ASNCLP session.
```

```
QUIT;
```

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

```
# 1 Setting the environment.
# The SET OUTPUT command creates two SQL scripts: qcapqmap.sql, which adds
# definitions for the queue map to the Q Capture control tables, and
# qappmap.sql, which adds definitions for the queue map to the Q Apply
# control tables.
```

```
ASNCLP SESSION SET TO Q REPLICATION;
SET LOG "rqmap.err";
SET SERVER CAPTURE TO DBALIAS SAMPLE ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE ASN1;
SET SERVER TARGET TO DB TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET APPLY SCHEMA ASN1;
SET OUTPUT CAPTURE SCRIPT "qcapmap.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 REPLQMAP SAMPLE_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 (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 I) using
# the EXPORT and IMPORT utilities (LOAD TYPE 2).

```

```

CREATE QSUB USING REPLQMAP SAMPLE_ASN1_TO_TARGET_ASN1
(SUBNAME EMPLOYEE0001 EMPLOYEE OPTIONS HAS LOAD PHASE I
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 DBALIAS 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 3. SQL script files that are created by the sample ASNCLP scripts

Output file	Contains SQL to ...
qcapctrl.sql	Create Q Capture control tables
qappctrl.sql	Create Q Apply control tables
qcapqmap.sql	Insert definitions for a replication queue map into the Q Capture control tables
qappqmap.sql	Insert definitions for a replication queue map into the Q Apply control tables
qcapqsub.sql	Insert definitions for a Q subscription into the Q Capture control tables
qappqsub.sql	Insert definitions for a Q subscription into the Q Apply control tables

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 65 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 WebSphere MQ setup script generator 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;
SET LOG "qcontrol.err";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passw0rd";
SET QMANAGER "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, asnpwd.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 "asnserver.ini"
```

```
ID DB2ADMIN PASSWORD "passw0rd";
SET RUN SCRIPT NOW;
ALTER CAPTURE PARAMETERS QMGR asnqmgr RESTARTQ asnrestart ADMINQ asnadmin;

# 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 "rqmap.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 "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_ASN_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 "asnserver.ini"
ID CLASSICADMIN PASSWORD "passwd";
SET SERVER TARGET TO DBALIAS TARGET ID DB2ADMIN PASSWORD "passwd";
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 4. SQL script files that are created by the sample ASNCLP scripts

Output file	Description
classicctrl.sql	Creates Q Apply control tables
qappqmap.sql	Inserts definitions for a replication queue map into the Q Apply control tables
qappqsub.sql	Inserts definitions for a Q subscription into the Q Apply control tables

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 70 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 WebSphere MQ setup script generator 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 REPLICATION;
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 rqmblue1.sql, 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 "rqmblue1.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.
# 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
- 4** Specifying replication queue maps to connect the servers in both directions
- 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.
```

```
# 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.

ASNCLP SESSION SET TO Q REPLICATION;
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, bidirsubs.in.

LOAD MULTIDIR REPL SCRIPT "/home/files/asncpl/bidirsubs.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 5. SQL script files that are created by the sample ASNCLP scripts

Output file	Contains SQL to ...
SAMPLE.sql	Create Q Capture and Q Apply control tables at the SAMPLE database.
SAMPLE2.sql	Create Q Capture and Q Apply control tables at the SAMPLE2 database.
rqmred1.sql	Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE_RED_TO_SAMPLE2_BLUE.
rqmblue1.sql	Add definitions to the Q Apply control tables at SAMPLE2 for the replication queue map SAMPLE_RED_TO_SAMPLE2_BLUE.
rqmred2.sql	Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE2_BLUE_TO_SAMPLE_RED.
rqmblue2.sql	Add definitions to the Q Apply control tables at SAMPLE2 for the replication queue map SAMPLE2_BLUE_TO_SAMPLE_RED.
SAMPLE.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.
SAMPLE2.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE2.

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 75 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 WebSphere MQ setup script generator 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 "passw0rd";
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 "passw0rd";
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 SAMPLPEER ID DB2ADMIN PASSWORD "passw0rd";
SET CAPTURE SCHEMA SOURCE MAGENTA;
SET QMANAGER "QM2" FOR CAPTURE SCHEMA;
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "MAGENTA.QM2.RESTARTQ" ADMINQ "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
- 4 Specifying queue maps that connect the servers in both directions
- 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.
```

```
# 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.GREEN.EMPLOYEE);

# 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/asnc1p/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

Output file	Contains SQL to ...
SAMPLE.sql	Create Q Capture and Q Apply control tables at the SAMPLE database.
SAMPLPEER.sql	Create Q Capture and Q Apply control tables at the SAMPLPEER database.
rqmgreen1.sql	Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLE_GREEN_TO_SAMPLPEER_MAGENTA.
rqmmagenta1.sql	Add definitions to the Q Apply control tables at SAMPLPEER for the replication queue map SAMPLE_GREEN_TO_SAMPLPEER_MAGENTA.
rqmgreen2.sql	Add definitions to the Q Capture control tables at SAMPLE for the replication queue map SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN.
rqmmagenta2.sql	Add definitions to the Q Apply control tables at SAMPLPEER for the replication queue map SAMPLPEER_MAGENTA_TO_SAMPLE_GREEN.
SAMPLE.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.
SAMPLPEER.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLPEER.

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 80 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 WebSphere MQ setup script generator 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 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 REPLICATION;
SET OUTPUT MULTIDIR;
SET LOG "p2p3-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 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";

# 3 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

2 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 REPLICATION;
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 SAMPLE_GRAY_TO_SAMPLE2_BROWN USING
ADMINQ "BROWN.QM2.ADMINQ" RECVQ "BROWN.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.QM1.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_SAMPLE_GRAY 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 SAMPLE_GRAY_TO_SAMPLE3_YELLOW USING
ADMINQ "YELLOW.QM3.ADMINQ" RECVQ "YELLOW.QM1_TO_QM3.DATAQ"
SENDQ "GRAY.QM1_TO_QM3.DATAQ";

# 2 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.

```

# 2 Setting the subgroup

SET SUBGROUP "p2p3group";

# 3 Setting servers for the subgroup

SET SERVER MULTIDIR TO DBALIAS "SAMPLE";
SET SERVER MULTIDIR TO DBALIAS "SAMPLE2";
SET SERVER MULTIDIR TO DBALIAS "SAMPLE3";

# 4 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;

# 5 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/asnc1p/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 7. SQL script files that are created by the sample ASNCLP scripts

Output file	Contains SQL to ...
SAMPLE.sql	Create Q Capture and Q Apply control tables at the SAMPLE database.
SAMPLE2.sql	Create Q Capture and Q Apply control tables at the SAMPLE2 database.

Table 7. SQL script files that are created by the sample ASNCLP scripts (continued)

Output file	Contains SQL to ...
SAMPLE3.sql	Create Q Capture and Q Apply control tables at the SAMPLE23 database.
rqm1.sql	Define the queue map SAMPLE_GRAY_TO_SAMPLE2_BROWN
rqm2.sql	Define the queue map SAMPLE2_BROWN_TO_SAMPLE_GRAY
rqm3.sql	Define the queue map SAMPLE2_BROWN_TO_SAMPLE3_YELLOW
rqm4.sql	Define the queue map SAMPLE3_YELLOW_TO_SAMPLE2_BROWN
rqm5.sql	Define the queue map SAMPLE3_YELLOW_TO_SAMPLE_GRAY
rqm6.sql	Define the queue map SAMPLE_GRAY_TO_SAMPLE3_YELLOW
SAMPLE.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE.
SAMPLE2.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE2.
SAMPLE3.sql	Add Q subscription definitions to the Q Capture and Q Apply control tables at SAMPLE3.

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-repqlmap-qsub.log;

SET SERVER CAPTURE TO DBALIAS TESTCAP ID id1 PASSWORD "p1"
PROMOTE TO DBALIAS PRODCAP ID id1 PASSWORD "p1wd" SCHEMA ASN;

SET SERVER TARGET TO DBALIAS TESTAPP ID id1 PASSWORD "p1wd"
PROMOTE TO DBALIAS PRODAPP ID id1 PASSWORD "p1wd" SCHEMA ASN;

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

#These two statements will be put in the generated script
SET OUTPUT CAPTURE SCRIPT "promote_capture_repqlmap.sql";
SET OUTPUT TARGET SCRIPT "promote_target_repqlmap.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 REPLQMAP LIKE "qmap%";
```

The output of this script is another ASNCLP script that is named `replqlmap_qsub.in` which includes the command `SET RUN SCRIPT LATER`. Using `SET RUN SCRIPT LATER` lets you confirm or modify the script contents after it is generated and before running it. Change this `SET RUN` command to `SET RUN SCRIPT NOW` `STOP ON SQL ERROR ON` when you want to run the script.

Running this script by using `asnclp -f "asnclp_replqlmap.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 destinations: on server SAMPLE under schema ASN1 and on server TESTDB1 under schema BSN1. Create the control tables by 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.

```
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.

```
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 "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 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%";
```

Chapter 4. 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 59 and “Sample ASNCLP scripts for setting up unidirectional Q replication from a Classic data source” on page 62 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

If you want to ...	Use this command
Add a column to a Q subscription	ALTER ADD COLUMN command
Update the IBMQREP_CAPPARMS table when you replicate from a Classic source	“ALTER CAPTURE PARAMETERS command (Classic replication)” on page 88
Change a Q subscription	“ALTER QSUB command (unidirectional replication)” on page 91
Change a replication queue map	ALTER REPLQMAP command
Establish a session for Q replication	ASNCLP SESSION SET TO command
Create a Q subscription	“CREATE QSUB command (unidirectional replication)” on page 103
Create the control tables for the Q Capture and Q Apply programs	CREATE CONTROL TABLES FOR command
Create a replication queue map	CREATE REPLQMAP command
Delete a Q subscription	“DROP QSUB command (unidirectional Q replication)” on page 120
Drop the control tables for the Q Capture and Q Apply programs	DROP CONTROL TABLES ON command
Delete a replication queue map	DROP REPLQMAP command
List Q subscriptions	“LIST QSUB command (Q replication)” on page 121
List replication queue maps	“LIST REPLQMAP command (Q replication)” on page 124
List Q Apply schemas	LIST APPLY SCHEMA command
List Q Capture schemas	LIST CAPTURE SCHEMA command
Signal that a manual load of the target table is complete	LOAD DONE command
Promote a Q subscription	PROMOTE QSUB command
Promote a replication queue map	PROMOTE REPLQMAP command
<ul style="list-style-type: none"> • Specify whether to drop the target table when you delete a Q subscription • Specify whether to drop the table space when you drop the target table or control tables 	“SET DROP command (unidirectional replication)” on page 131
Set the Q Apply schema for all task commands	SET APPLY SCHEMA command

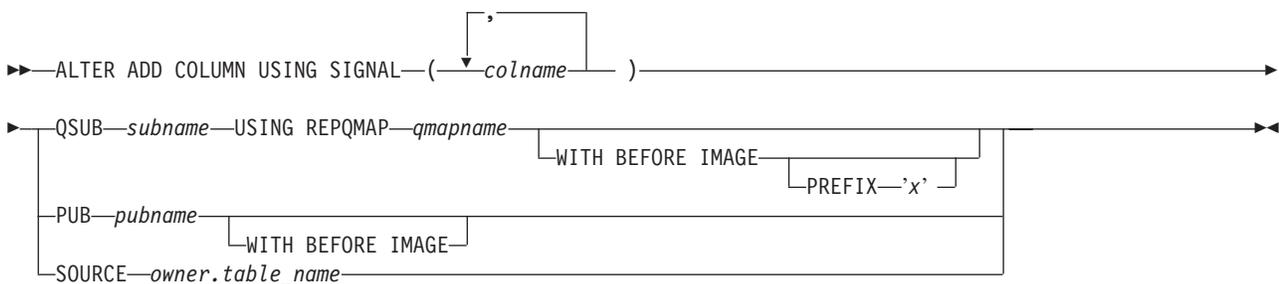
Table 8. ASNCLP commands for unidirectional Q replication (continued)

If you want to ...	Use this command
Set the Q Capture schema for all task commands	SET CAPTURE SCHEMA command
Define the log file for the ASNCLP program	SET LOG command
Define output files that contain SQL statements to set up unidirectional Q replication	SET OUTPUT command
Specify custom parameters for database objects to be created implicitly	SET PROFILE command
Set the WebSphere MQ queue manager	SET QMANAGER command
Specify whether to automatically run each task command from an input file before the ASNCLP program processes the next task command	SET RUN SCRIPT command
Specify the Q Capture server or Q Apply server to use in the ASNCLP session for unidirectional replication.	SET SERVER command
Enable and disable the trace for the ASNCLP commands	SET TRACE command
Display the environment set during the session	SHOW SET ENV command
Start a Q subscription	START QSUB command
Stop a Q subscription	STOP QSUB command
Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions	VALIDATE WSMQ ENVIRONMENT FOR command
Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map	VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command

ALTER ADD COLUMN command

Use the ALTER ADD COLUMN command to add a column to a Q subscription or publication.

Syntax



Parameters

colname

Specifies one or more columns (separated by a comma) to add to the definition of the active Q subscription or publication.

QSUB *subname*

Specifies the name of the Q subscription.

WITH BEFORE IMAGE

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, DB2 uses the default, X. If this prefix generates invalid names then other letters will be used beginning with Y until valid names are generated.

USING REPMAP *qmapname*

Specifies the name of the replication queue map used by the Q subscription.

PUB *pubname*

Specifies the name of the publication.

SOURCE *owner.table_name*

Specifies that the columns are added to all of the subscriptions and publications that subscribe to the source table.

Usage notes

- The column needs to exist in the source table already and should not be part of any existing Q subscription or publication.
- The Q subscription or publication must be active.
- The column must be nullable or have a default value on the source table.
- The column name on the target table will be implicitly named the same as the column name on the source table.
- For LONG VARCHAR or GRAPHIC types, the DATA CHANGES INCLUDE VARCHAR COLUMNS option must be enabled. VARCHAR COLUMNS are variable length character columns. The DATA CHANGES INCLUDE VARCHAR COLUMNS is an option set on the source table by altering the table's attributes using SQL.
- There is a limit of 20 columns that can be inserted into the statement.
- This command works for unidirectional replication and event publishing.

Example 1

To alter a Q subscription by adding columns PHONE and ADDRESS to the EMPLOYEE0001 Q subscription:

```
ALTER ADD COLUMN USING SIGNAL (PHONE, ADDRESS) QSUB EMPLOYEE0001
USING REPMAP SAMPLE_ASN_TO_TARGETDB_ASN
```

Example 2

To add the PHONE, ADDRESS, and EMAIL columns to all subscriptions and publications that subscribe to the EMPLOYEE table.

```
ALTER ADD COLUMN USING SIGNAL (PHONE, ADDRESS, EMAIL) SOURCE DB2ADMIN.EMPLOYEEE
```

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—

|            |
|------------|
| set—"name" |
| is null    |

—▶▶
```

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 REPLQMAP command

Use the ALTER REPLQMAP command to customize attributes for an existing replication queue map. This command applies to Q replication and Classic replication.

Syntax

```
▶▶ ALTER REPLQMAP qmapname USING | options |
```

options:

```
| [DESC "description" ] [ADMINQ "adminqname" ] [RECVQ "recvqname" ] [SENDQ "sendqname" ]  
▶ [NUM APPLY AGENTS num ] [MAXAGENTS CORRELID num ] [MEMORY LIMIT limit ]  
▶ [ERROR ACTION  S  Q ] [HEARTBEAT INTERVAL interval ] [MAX MESSAGE SIZE size ]
```

Parameters

qmapname

Specifies the name of the replication queue map.

DESC "*description*"

Specifies the description of the replication queue map.

ADMINQ "*adminqname*"

Specifies the name of the administration queue at the Q Apply server.

Note: If the Q Capture or Classic capture components share a single queue manager with the Q Apply programs, they can share an administration queue.

RECVQ "*recvqname*"

Specifies the name of the receive queue that is used by the Q Apply program.

SENDQ *"sendqname"*

Specifies the name of the send queue that is used by the Q Capture program or Classic capture components.

NUM APPLY AGENTS *num*

Specifies the number of threads that are used to concurrently apply transactions from the specified receive queue.

MAXAGENTS CORRELID*num*

z/OS Specifies that number of threads that are used for concurrently applying transactions from the specified receive queue with the same *correlation ID*. The correlation ID identifies all transactions that were started from the same z/OS job on the Q Capture server.

The value for the **MAXAGENTS CORRELID** parameter cannot be greater than the value for the **NUM APPLY AGENTS** parameter. If

MAXAGENTS_CORRELID value is 1, the transactions will be applied one at a time. If the value is greater than one, for example 4, four agents will apply transactions with the same correlation ID in parallel. If the value is 0, transactions are applied in parallel by using the total number of threads specified by the **NUM APPLY AGENTS** parameter.

MEMORY LIMIT *limit*

Specifies the maximum number of megabytes that are used per receive queue to buffer incoming transactions.

ERROR ACTION

The action that the Q Capture program takes 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.

S The Q Capture program or the capture components stop when they detect an error on this queue.

Q The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues. This value is not supported for Classic replication.

HEARTBEAT INTERVAL *interval*

Specifies the interval (in seconds) between heartbeat messages that are sent by the Q Capture program or Classic capture components to the Q Apply program when there are no transactions to publish.

MAX MESSAGE SIZE *size*

Specifies the maximum size (in kilobytes) of the buffer that is used for sending messages over the send queue. The size of the buffer must not be larger than the maximum message length (**MAXMSGL**) that is defined for the send queue.

Example 1

The following command alters the **SAMPLE_ASN1_TO_TARGETDB_ASN1** replication queue map, sets the threads to 4, and invalidates all of the Q subscriptions that use the send queue for this replication queue map if an error occurs.

```
ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO SAMPLE;
SET CAPTURE SCHEMA ASN1;
SET SERVER TARGET TO TARGETDB
SET APPLY SCHEMA ASN1;
ALTER REPLQMAP SAMPLe_ASN1_TO_TARGETDB_ASN1 USING NUM APPLY AGENTS 4
ERROR ACTION I;
```

Example 2

The following command alters the CLASSIC_ASN_TO_TARGETDB_ASN1 replication queue map, sets the threads to 4, sets the maximum memory limit to 10 megabytes, stops the Classic capture components if an error occurs, sets the heartbeat interval to 4, and sets the maximum buffer size to 5 kilobytes.

```
ASNCLP SESSION SET TO Q REPLICATION;  
SET OUTPUT TARGET SCRIPT "replapp.sql";  
SET LOG "qmap.err";  
SET SERVER CAPTURE TO CONFIG SERVER server1 FILE "asnservers.ini"  
    ID username PASSWORD "passw1rd";  
SET SERVER TARGET TO DB TARGETDB;  
SET APPLY SCHEMA ASN1;  
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;  
ALTER REPLQMAP CLASSIC_ASN_TO_TARGETDB_ASN1 USING NUM APPLY AGENTS 4  
    MEMORY LIMIT 10 ERROR ACTION S HEARTBEAT INTERVAL 4 MAX MESSAGE SIZE 5;
```

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 ] [ add-cols-clause ] ]
```

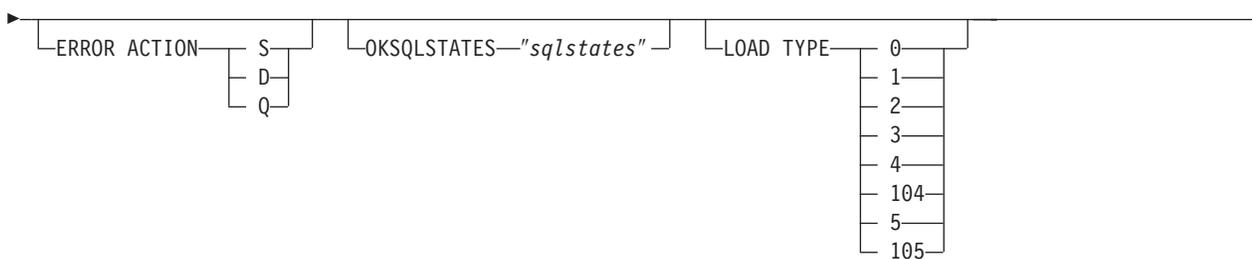
action:

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

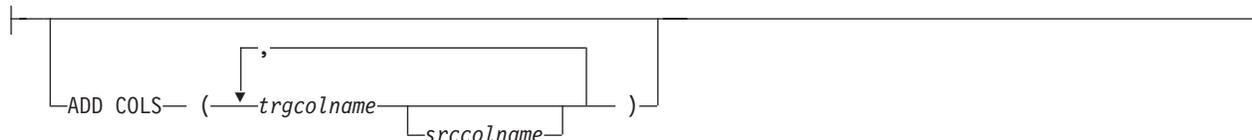
other-opt-clause:

```
[ SEARCH CONDITION "search_condition" ] [ ALL CHANGED ROWS [ N ] [ Y ] ]
```

```
▶ [ HAS LOAD PHASE [ N ] [ I ] [ E ] ] [ SUPPRESS DELETES [ N ] [ Y ] ] [ CONFLICT ACTION [ I ] [ F ] [ D ] [ S ] [ Q ] ]
```



add-cols-clause:



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 program 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 replqmap10 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 For a DB2 source, disable subscription and notify Q Capture. For a Classic source, disable subscription and notify the Classic capture components.

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 method of loading the target table with data from the source.

0 Choose the best type automatically. Not valid for Classic sources.

1 Use LOAD from CURSOR only. Not valid for Classic sources or federated targets.

2 Use SELECT/IMPORT only. Not valid for Classic sources.

- 3 Use SELECT/LOAD only. Not valid for Classic sources or for federated targets.
- 4 Select from Classic sources and use the DB2 LOAD utility with the replace option.

104

Select from a Classic source and use the LOAD utility with resume option (z/OS targets) or insert option (Linux, UNIX, and Windows targets). Use this load option when an existing Q subscription writes to the same target table as this Q subscription with a replace option (LOAD TYPE 4 or 5), to prevent this Q subscription from overwriting the table data from the earlier Q subscription.

- 5 **Linux, UNIX, and Windows targets:** Select from Classic sources and use the DB2 IMPORT utility with the replace option. Use this load option when the source code page differs from the target code page. The DB2 IMPORT utility invoked by this option converts code pages.

105

Linux, UNIX, and Windows targets: Select from Classic sources and use the DB2 IMPORT utility with insert option. Use this load option when the source code page differs from the target code page, and when an existing Q subscription writes to the same target table as this Q subscription with a replace option (LOAD TYPE 4 or 5). The DB2 IMPORT utility invoked by this option converts code pages, and the insert option prevents this Q subscription from overwriting the table data from the earlier Q subscription.

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 - Changing selected properties

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 I
SUPPRESS DELETES N CONFLICT ACTION F ERROR ACTION Q LOAD TYPE 1
```

This example is valid only with DB2 sources.

Example - Adding columns

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)
```

This example does not apply to Classic replication because with it you must replicate all columns. You cannot add columns.

Example - Creating a registration for SQL replication

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 - Deleting a registration for SQL replication

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;

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 DROP SQL REGISTRATION;
```

ASNCLP SESSION SET TO command

Use the ASNCLP SESSION SET TO command to establish an ASNCLP session for Q replication to either relational or Classic data sources.

Syntax

▶▶—ASNCLP SESSION SET TO—Q REPLICATION—◀◀

Parameters

Q REPLICATION

Specify to set the ASNCLP session to Q replication. This ASNCLP session only accepts Q replication syntax. Use this parameter when you are connecting to either relational or Classic sources.

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.
- You can only issue commands that apply to the type of replication that you specify.

Example 1

To set the ASNCLP session to Q replication:

```
ASNCLP SESSION SET TO Q REPLICATION
```

CREATE CONTROL TABLES FOR command

Use the CREATE CONTROL TABLES FOR command to set up Q Capture and Q Apply control tables. For event publishing, Q Apply control tables are not needed.

For bidirectional and peer-to-peer replication, run the SET MULTIDIR SCHEMA command before you use this command. The Q Capture and Q Apply programs must use the same schema on each server.

In Classic replication, the control tables for the Classic capture components are created by using the Classic Data Architect.

Syntax

```
▶▶ CREATE CONTROL TABLES FOR { CAPTURE SERVER | APPLY SERVER } USING capparms-clause |  
                                USING applyparms-clause
```

```
▶ { ZOS INDEX | zos-idx-clause }
```

capparms-clause:

```
▶ { IN { ZOS | ZOS- clause | ZOS- clause | UW | UW- clause } | RESTARTQ "rstqname" | ADMINQ "admname" }
```

```
▶ { STARTMODE { WARM | WARM | COLD | WARM } | MEMORY LIMIT limit | AUTOSTOP { N | Y } }
```

```
▶ { MONITOR INTERVAL interval | MONITOR LIMIT monlimit | TRACE LIMIT trclimit }
```

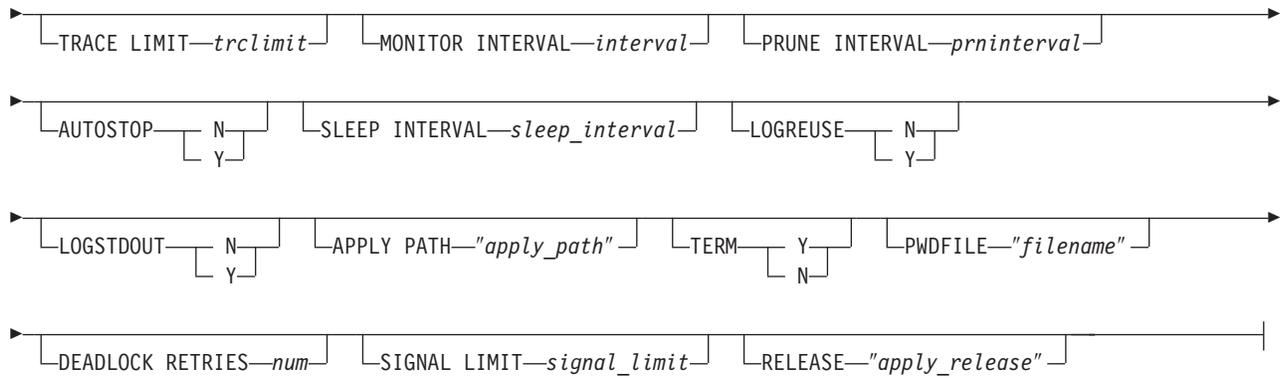
```
▶ { SIGNAL LIMIT siglimit | PRUNE INTERVAL prninterval | SLEEP INTERVAL sleepinterval }
```

```
▶ { LOGREUSE { N | Y } | LOGSTDOUT { N | Y } | TERM { Y | N } | CAPTURE PATH "capture_path" }
```

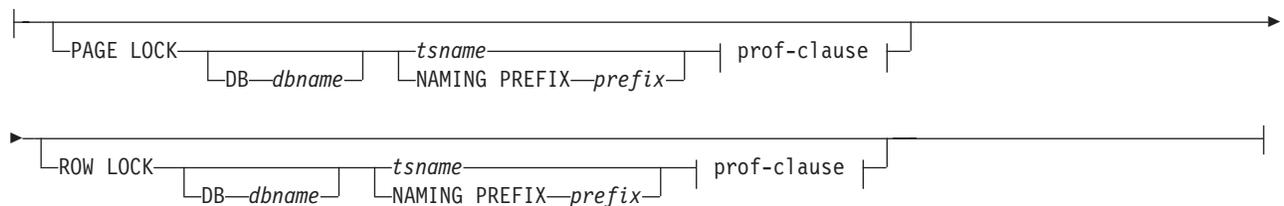
```
▶ { RELEASE "capture_release" }
```

applyparms-clause:

```
▶ { IN { ZOS | ZOS- clause | ZOS- clause | UW | UW- clause | FEDERATED | FEDERATED- clause } | MONITOR LIMIT monitor_limit }
```



zos-ts-clause:



uw-ts-clause:



fed-ts-clause:



prof-clause:



zos-idx-clause:



Parameters

CAPTURE SERVER

Specify to create Q Capture control tables.

APPLY SERVER

Specify to create Q Apply control tables.

capparms-clause:

ZOS

Specifies a z/OS system on which to create Q Capture control tables.

UW

Specifies a Linux, UNIX, or Windows system on which to create Q Capture control tables.

RESTARTQ "rstqname"

Specifies the restart queue that the Q Capture program uses.

ADMINQ "admqname"

Specifies the administration queue that the Q Capture program uses.

STARTMODE

Specifies what kind of start the Q Capture program will perform.

WARMSI

Specify for the Q Capture program to perform a warm start. If the Q Capture program is starting for the first time, it will perform a cold start.

COLD

Specify for the Q Capture program to perform a cold start.

WARMNS

Specify for the Q Capture program to attempt a warm start if information is available. If the information is not available, the Q Capture program will stop.

MEMORY LIMIT *limit*

Specifies the maximum amount (in MB) of memory that the Q Capture program can use to build transactions.

AUTOSTOP

N The Q Capture or Q Apply program does not stop after it reaches the end of the active log and finds no transactions.

Y The Q Capture or Q Apply program stops after it reaches the end of the active log and finds no transactions.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Capture program inserts rows into the IBMQREP_CAPMON table.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPMON and IBMQREP_CAPQMON tables before it becomes eligible for pruning. All rows in these tables that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

SIGNAL LIMIT *siglimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_SIGNAL table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_CAPMON, IBMQREP_CAPQMON, IBMQREP_CAPTRACE, and IBMQREP_SIGNAL tables are pruned.

SLEEP INTERVAL *sleepinterval*

Specifies the number of milliseconds that the Q Capture program sleeps when it finishes processing the active log and determines that the buffer is empty.

LOGREUSE

N The Q Capture program appends messages to the log file, even after the Q Capture program restarts.

Y The Q Capture program reuses the log file by first truncating the current log file and then starting a new log when the Q Capture program restarts.

LOGSTDOUT

N The Q Capture program only sends messages to the log file.

Y The Q Capture program sends messages to both the log file and the standard output (stdout).

TERM

Y The Q Capture program terminates if DB2 is quiesced or stops. This value is the default.

N The Q Capture program continues running if DB2 is quiesced or stops.

CAPTURE_PATH *"capture_path"*

Specifies the location of the work files that the Q Capture program uses. On z/OS systems, the location can be an MVS™ data set high-level qualifier with // . The default is NULL.

Linux UNIX Windows RELEASE *"capture_release"*

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks ("). Specifying the release level enables newer replication and publishing function on an older DB2.

appparms-clause:

ZOS

Specifies a z/OS system in which Q Apply control tables are created.

UW

Specifies a Linux, UNIX, or Windows system in which Q Apply control tables are created.

FEDERATED

Specifies a federated target, on which Q Apply control tables are created in an Oracle, Sybase, Informix, Microsoft SQL Server, or Teradata database, and nicknames are created for these control tables in the Q Apply server. Some control tables are created in the Q Apply server.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_APPLYMON table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the

IBMQREP_APPLYTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Apply program inserts rows into the IBMQREP_APPLYMON table.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_APPLYMON and IBMQREP_APPLYTRACE tables are pruned.

AUTOSTOP

N The Q Apply program does not stop after all queues are emptied once.

Y The Q Apply program stops after all queues are emptied once.

LOGREUSE

N The Q Apply program appends messages to the log file, even after the Q Apply program is restarted.

Y The Q Apply program reuses the log file by first truncating the current log file and then starting a new log when the Q Apply program is restarted.

LOGSTDOUT

N The Q Apply program sends messages only to the log file.

Y The Q Apply program sends messages to the log file and the standard output (stdout).

APPLY PATH "*apply_path*"

Specifies the location of the work files the Q Apply program uses. The default path is the directory where the asnqapp command was run.

TERM

Y The Q Apply program stops if DB2 is quiesced or stops.

N The Q Apply program continues running if DB2 is quiesced or stops.

PWDFILE "*filename*"

Specifies the name of the password file.

DEADLOCK RETRIES *num*

Specifies the number of retries for SQL deadlock errors.

Linux UNIX Windows RELEASE "*apply_release*"

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks ("). Specifying the release level enables newer replication and publishing function on an older DB2.

zos-ts-clause:

PAGE LOCK

Specify for replication control tables that require page-level locking.

ROW LOCK

Specify for replication control tables that require row-level locking.

DB *dbname*

Specifies the name of the database that contains the table space where the control tables will be created.

tsname

Specifies the name of the table space for the z/OS control tables.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

uw-ts-clause:

TBSPACE

tsname

Specifies the name of the table space that is used for the control tables on Linux, UNIX, or Windows.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

fed-ts-clause:

TBSPACE *tsname*

Specifies the name of an existing Oracle table space, Sybase segment, Informix dbspace, or Microsoft SQL Server file group that is used for the control tables. This parameter is not applicable for Teradata targets.

RMT SCHEMA

The remote schema that the Q Apply program uses to create control tables on the non-DB2 database. The default is the remote authorization ID.

CREATE

Specify to create a table space. When this parameter is used without the USING PROFILE keyword, the table space is assumed to exist and the control tables are created in this table space.

USING PROFILE *pname*

Specifies the name of a profile to use to customize the table space attributes.

Example 1

To create Q Apply control tables and to specify a monitor limit of 3 minutes and a trace limit of 9 minutes:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING MONITOR LIMIT 3 TRACE LIMIT 9
```

Example 2

To create Q Capture control tables:

```
CREATE CONTROL TABLES FOR CAPTURE SERVER USING  
RESTARTQ "ASN1.QM1.RESTARTQ" ADMINQ "ASN1.QM1.ADMINQ"
```

Example 3

To create Q Apply control tables for replication to an Oracle target with a remote authorization ID of ORACLE_ID:

```
CREATE CONTROL TABLES FOR APPLY SERVER IN FEDERATED RMT SCHEMA ORACLE_ID
```

Example 4

To create Version 9.7 Q Apply control tables on a DB2 Version 9.1 database:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING RELEASE "9.7"
```

CREATE REPLQMAP command

Use the CREATE REPLQMAP command to create a replication queue map for Q subscriptions.

Syntax

```
▶▶ CREATE REPLQMAP qmapname [DESC "description"] USING ADMINQ "adminqname"
▶ RECVQ "recvqname" SENDQ "sendqname" [NUM APPLY AGENTS num] [MAXAGENTS CORRELID num]
▶ [MEMORY LIMIT limit] [ERROR ACTION {S|Q}] [HEARTBEAT INTERVAL interval]
▶ [MAX MESSAGE SIZE size]
```

Parameters

qmapname

Specifies the name of the replication queue map.

DESC "*description*"

Specifies the description of the replication queue map.

ADMINQ "*adminqname*"

Specifies the name of the administration queue at the Q Apply server.

Note: If the Q Capture or the Classic capture components share a single queue manager with the Q Apply program, the programs can share an administration queue.

RECVQ "*recvqname*"

Specifies the name of the receive queue that is used by the Q Apply program.

SENDQ "*sendqname*"

Specifies the name of the send queue that is used by the Q Capture program (for relational sources) or the capture components.

NUM APPLY AGENTS *num*

Specifies the number of threads that are used for concurrently applying transactions from the specified receive queue.

MAXAGENTS CORRELID *num*

z/OS Specifies that number of threads that are used for concurrently applying transactions from the specified receive queue with the same *correlation ID*. The correlation ID identifies all transactions that were started from the same z/OS job on the Q Capture server.

The value for the **MAXAGENTS CORRELID** parameter cannot be greater than the value for the **NUM APPLY AGENTS** parameter. If

MAXAGENTS_CORRELID value is 1, the transactions will be applied one at a time. If the value is greater than one, for example 4, four agents will apply transactions with the same correlation ID in parallel. If the value is 0, transactions are applied in parallel by using the total number of threads specified by the **NUM APPLY AGENTS** parameter.

MEMORY LIMIT *limit*

Specifies the maximum number of megabytes that are used per receive queue for buffering incoming transactions.

ERROR ACTION

The action that the Q Capture program takes 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.

S The Q Capture program or the capture components stop when they detect an error on this queue.

Q The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues. This value is not supported for Classic replication.

HEARTBEAT INTERVAL *interval*

Specifies the interval (in seconds) between heartbeat messages that are sent from the Q Capture program or the capture components to the Q Apply program when there are no transactions to publish.

MAX MESSAGE SIZE *size*

Specifies the maximum size (in kilobytes) of the buffer that is used for sending messages over the send queue.

Example 1

To create a replication queue map SAMPLE_ASN1_TO_TARGETDB_ASN1 from a relational source:

```
CREATE REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1 USING ADMINQ "ASN1.QM1.ADMINQ"
RECVQ "ASN1.QM1_TO_QM2.DATAQ" SENDQ "ASN1.QM1_TO_QM2.DATAQ"
```

Example 2

To create a replication queue map CLASSIC_ASN_TO_TARGETDB_ASN1 from a Classic source:

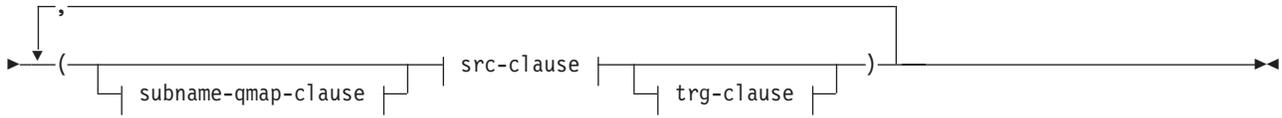
```
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE classic.ini ID id1 PASSWORD pwd1
SET SERVER TARGET TO DB ASN1
SET RUN SCRIPT NOW STOP ON SQL ERROR ON
CREATE REPLQMAP CLASSIC1_ASN_TO_TARGETDB_ASN1 USING ADMINQ "ASN1.QM1.ADMINQ"
RECVQ "CLASSIC1.QM1_TO_QM2.DATAQ" SENDQ "CLASSIC1.QM1_TO_QM2.DATAQ"
```

CREATE QSUB command (unidirectional replication)

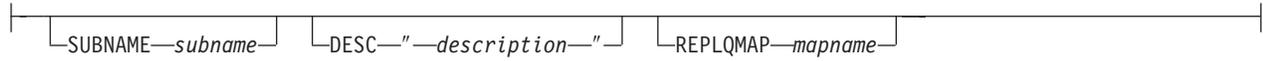
Use the CREATE QSUB command to create a Q subscription that maps a source table to a target table. For Classic replication, a Q subscription maps a source table or view in the Classic metadata catalog to a target table.

Syntax

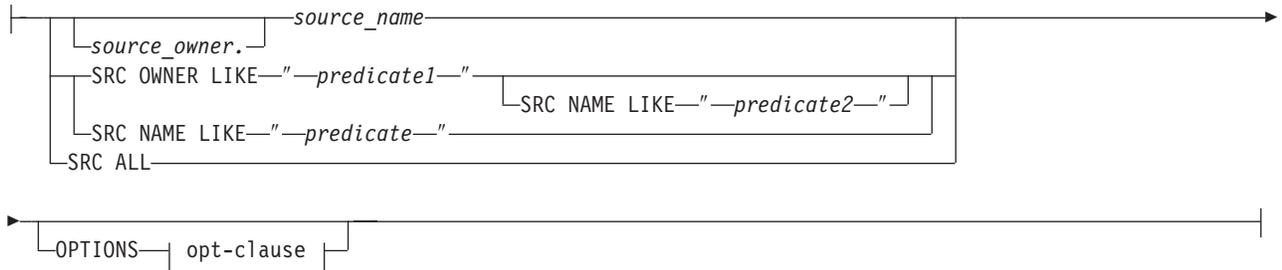
```
►► CREATE QSUB [SUBTYPE U] USING REPLQMAP mapname
```



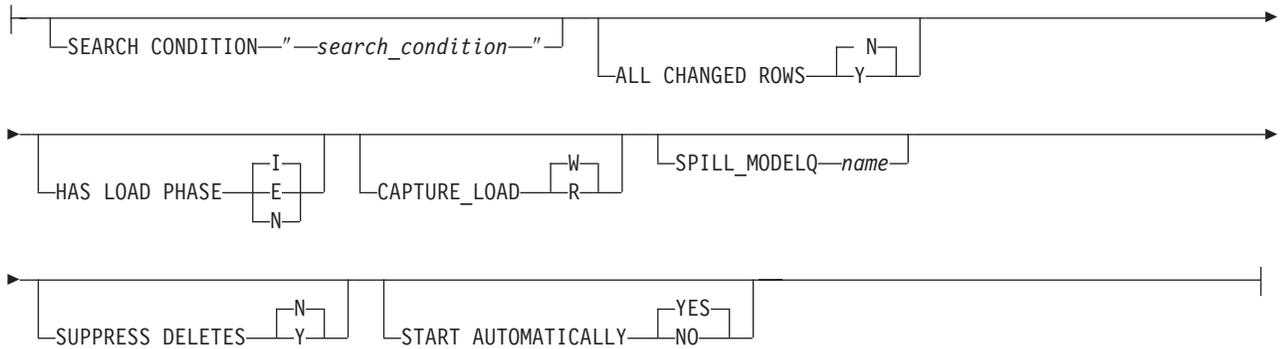
subname-qmap-clause:



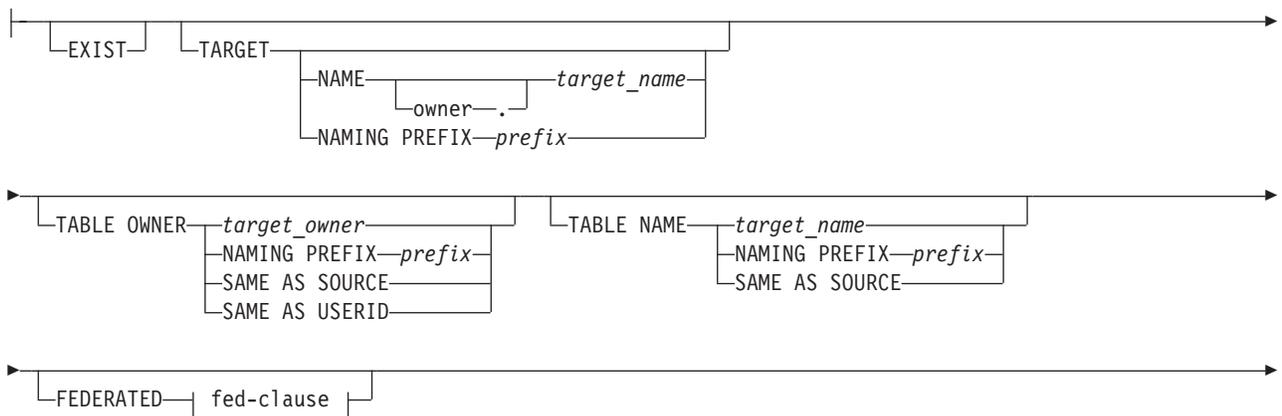
src-clause:

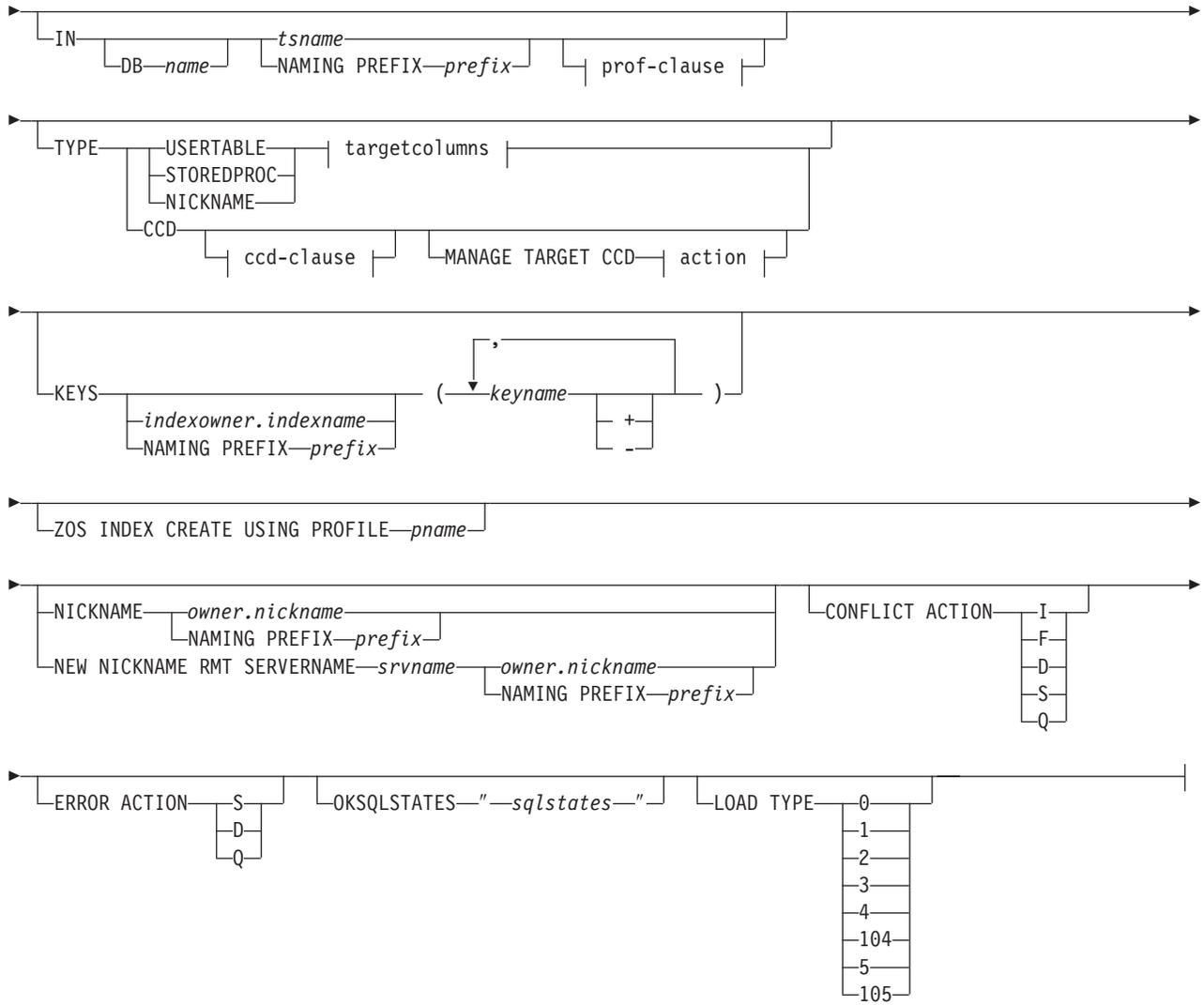


opt-clause:

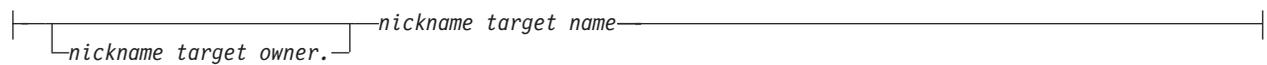


trg-clause:

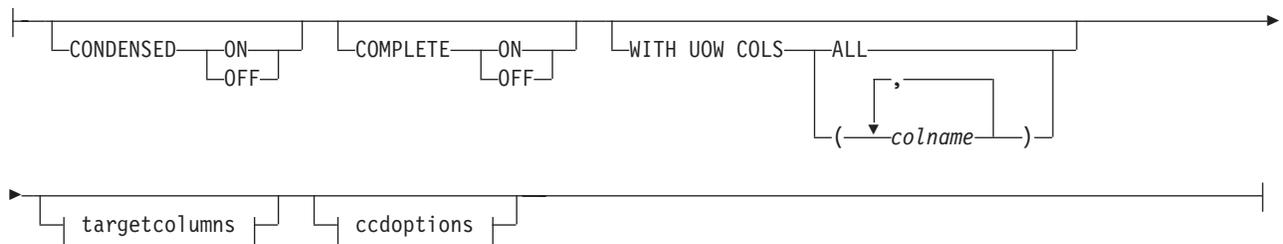




fed-clause:



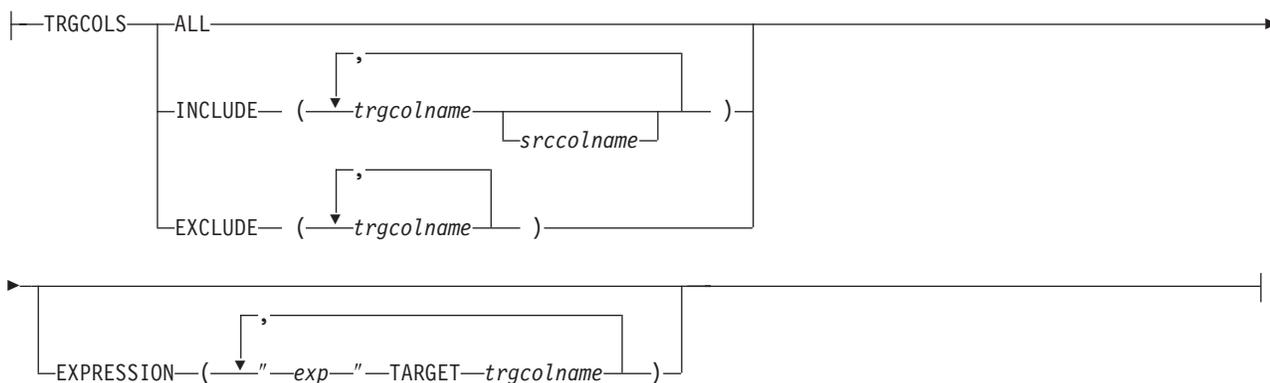
ccd-clause:



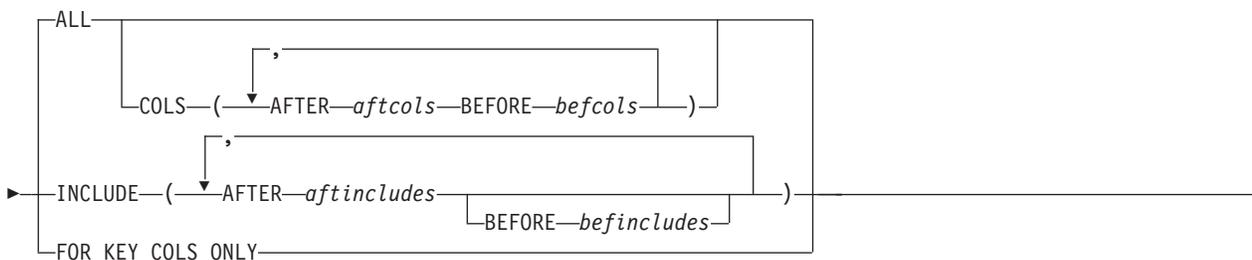
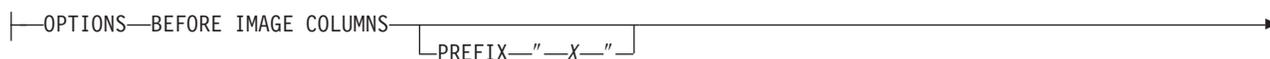
prof-clause:



targetcolumns:



ccdoptions:



action:



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-qmap-clause

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 mysubname ALLTYPE1 OPTIONS SEARCH CONDITION
"WHERE :MYKEY > 1000")
```

ALL CHANGED ROWS

Specifies the data sending option.

N (default)

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.

I (default)

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.

N No load phase at the target.

CAPTURE_LOAD

Specifies the action that the Q Capture program takes when the recovery log shows that a load operation that uses the DB2 LOAD utility occurred at the source table.

W (default)

Q Capture issues a warning message after the load completes.

R Q Capture issues a warning message and then stops and starts the Q subscription for the source table, prompting a load of the target table if one is specified for the Q subscription.

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 (default)

Send deleted rows.

Y Do not send deleted rows.

START AUTOMATICALLY

Specifies how to start the Q subscription, which is represented by the State column in the IBMQREP_SUBS table. The State column controls whether the subscription is automatically started after starting or reinitializing the Q Capture program (subscription state N), or that the subscription must be started manually by inserting a command in the IBMQREP_SIGNAL table (subscription state I).

YES

The Q subscription is started automatically (subscription state value of N). This is the default.

NO

The Q subscription must be started manually (subscription state value of I).

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 method of loading the target table with data from the source.

0 Choose the best type automatically. Not valid for Classic sources.

1 Use LOAD from CURSOR only. Not valid for Classic sources or federated targets.

2 Use SELECT/IMPORT only. Not valid for Classic sources.

3 Use SELECT/LOAD only. Not valid for Classic sources or for federated targets.

4 Select from Classic sources and use the DB2 LOAD utility with the replace option.

104

Select from a Classic source and use the LOAD utility with resume option (z/OS targets) or insert option (Linux, UNIX, and Windows targets). Use this load option when an existing Q subscription writes to the same target table as this Q subscription with a replace option (LOAD TYPE 4 or 5), to prevent this Q subscription from overwriting the table data from the earlier Q subscription.

5 Linux, UNIX, and Windows targets: Select from Classic sources and use the DB2 IMPORT utility with the replace option. Use this load option when the source code page differs from the target code page. The DB2 IMPORT utility invoked by this option converts code pages.

Linux, UNIX, and Windows targets: Select from Classic sources and use the DB2 IMPORT utility with insert option. Use this load option when the source code page differs from the target code page, and when an existing Q subscription writes to the same target table as this Q subscription with a replace option (LOAD TYPE 4 or 5). The DB2 IMPORT utility invoked by this option converts code pages, and the insert option prevents this Q subscription from overwriting the table data from the earlier Q subscription.

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.

targetcolumns

TRGCOLS

ALL

Specify to replicate all columns from the source table.

INCLUDE

Specifies the replicated columns in the target table. If the target table does not exist, specifies the column definitions 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 sub9 dpropr64.srctable
EXIST TARGET NAME dpropr64.trgtable
TRGCOLS INCLUDE (one, two))
```

srccolname

Specify to define a target table column that uses the properties of the specified source column, but when the target column has a different name than the source column. In the following example, the target table defines two columns *target_one* and *target_two* based on the properties of corresponding columns *one* and *two* in the source table:

```
CREATE QSUB SUBTYPE U USING REPLQMAP replqmap9
(SUBNAME sub9 dpropr64.srctable
EXIST TARGET NAME dpropr64.trgtable
TRGCOLS INCLUDE (target_one one, target_two two))
```

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))
```

trgcolname

Specify to exclude from the target table a column definition that uses the provided name and the properties of a source column with the same name.

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 value of is used. If this prefix generates invalid names, other letters will be used beginning with the letter 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 required. 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 EC06V71A DBNAME stlec1 ID ADMF001 password "xx";
SET SERVER TARGET to dbALIAS EC06V71A 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 "UITRGTS" FOR OBJECTS TARGET INDEX OPTIONS ZOS
  BUFFERPOOL BP1 STOGROUP "DPROSTGQ"
  PRIQTY ABSOLUTE 100 SECQTY ABSOLUTE 50;
SET PROFILE "UTRGTS" FOR OBJECT TARGET TABLESPACE OPTIONS ZOS
  DB "JUTRGDB"
  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.

```
ASNCLP SESSION SET TO Q REPLICATION;
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.

```
ASNCLP SESSION SET TO Q REPLICATION;
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
  NICKNAME REPLDBA.NICK_T2 LOAD TYPE 1);
```

Example 4

This example demonstrates the use of a naming prefix for the target table (XNEW) and table space 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 table space profile (USING PROFILE UTRGTS) for the target table space 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 table space (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 115. It creates a Q subscription for unidirectional replication from a DB2 source that uses the replication queue map SAMPLE_ASN1_TO_TARGETDB_ASN1 and

specifies that the Q Apply program loads the target tables with 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 1
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 of Y.

```
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 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 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 EXIST 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 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 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 TESTCCCDEXIST 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 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 | QAPPLY]
[SCHEMA—schema] [SERVER—servername] [dbparms]
```

dbparms-clause:

```
[DB—dbalias | DBALIAS—aliasname | DBNAME—dbname | ID—userid | PASSWORD—pwd]
[CONFIG SERVER—servername | FILE—filename]
```

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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 the Classic source that the ASNCLP program connects to. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file. You cannot use this parameter if you are using the **TARGET** parameter.

FILE *filename*

Specifies the complete path and file name to the ASNCLP 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 - list by Q Capture schema

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 - list by Classic server schema

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 APPLY SCHEMA command

You can use the LIST APPLY SCHEMA command to list the Q Apply schemas for a specified server.

Syntax

```
▶▶—LIST APPLY SCHEMA—┬──SERVER──┴──dbparms──┬──▶▶
```

dbparms-clause:

```
┬──DBALIAS—aliasname──┬──DBNAME—dbname──┬──ID—userid──┬──PASSWORD—pwd──┬──  
└──CONFIG SERVER—servername──┬──FILE—filename──┘
```

Parameters

dbparms-clause:

SERVER

Specifies the server that contains the schemas to be listed.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

Example

To list the Q Capture schema on server SAMPLE:

```
LIST CAPTURE SCHEMA SERVER DBALIAS SAMPLE ID id1 PASSWORD "passwd!";
```

LIST CAPTURE SCHEMA command

You can use the LIST CAPTURE SCHEMA command to list the Q Capture schemas for a specified server.

Syntax

```
▶▶ LIST CAPTURE SCHEMA [SERVER servername] [dbparms dbparms]
```

dbparms-clause:

```
[DBALIAS aliasname] [DBNAME dbname] [ID userid] [PASSWORD pwd]  
[CONFIG SERVER servername] [FILE filename]
```

Parameters

dbparms-clause:

SERVER

Specifies the server that contains the schemas to be listed.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

Example

To list the Q Capture schema on server SAMPLE:

```
LIST CAPTURE SCHEMA SERVER DBALIAS SAMPLE ID id1 PASSWORD "passwd!";
```

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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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";
```

LOAD DONE command

Use the LOAD DONE command to inform the Q Capture program or the Classic capture components that the target table is loaded. Issue the LOAD DONE command only if you are doing a manual load. If the Q Apply program is doing the load, this signal is not necessary.

Syntax

```
LOAD DONE QSUB SUBNAME subname
                FOR SUBNAME LIKE "%text%"
                CAP SERVER OPTIONS classic-opt-clause
```

classic-opt-clause:

```
DBALIAS aliasname DBNAME dbname ID userid PASSWORD pwd
CONFIG SERVER servername FILE filename
CAPSCHEMA schema
```

Parameters

SUBNAME *subname*

Specifies the name of the Q subscription for the LOADDONE signal.

FOR SUBNAME LIKE "%text%"

Specify to signal that the load is done for Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
LOAD DONE QSUB FOR SUBNAME LIKE "%table%"
```

CAP SERVER OPTIONS

Specifies additional parameters when you issue the LOAD DONE command in immediate execution mode.

classic-opt-clause: These parameters only work with Classic sources.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

ID *userid*

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

PASSWORD *pwd*

Specifies the password to use to connect to the source database.

CAPSCHEMA *schema*

Specifies the schema of the control tables of the Classic source.

Parameters

SUBNAME *subname*

Specifies one or more Q subscription names to promote. Separate multiple Q subscription names with commas.

LIKE "*predicate*"

Specifies a list of Q subscription names to promote that match the predicate.

REPLQMAP *replqmap*

Specifies one or more replication queue maps. Separate multiple map names with commas. All Q subscriptions that use the specified map or maps are promoted.

LIKE "*predicate*"

Specifies a list replication queue maps that match the predicate. All Q subscriptions that use the matching maps are promoted.

AT NODE *num*

Specifies the node number of the source configuration to be promoted. Default value is 1 for NODE 1. This option is ignored when you promote unidirectional Q subscriptions.

A "node" is a paired Q Capture/Q Apply schema at a server that is participating in bidirectional or peer-to-peer replication. For example you could have three physical machines that were involved in peer-to-peer replication, each with a DB2 database. Within each database are one or more Q Capture/Q Apply programs and their control tables that are identified by a schema. The paired schema represents a "node" in a three-way peer-to-peer configuration

new-clause:

USING SOURCE SCHEMA *schema*

Specifies the source table schema.

USING TARGET SCHEMA *schema*

Specifies the target table schema. If the schema is not specified, the promoted definition uses the schema of the current target table.

USING REPLQMAP *newqmap*

Specifies the name of a new replication queue map that you want to use for the promoted Q subscriptions.

Example - matching a predicate

To promote all bidirectional Q subscriptions that match the predicate EMP at NODE 1:

```
PROMOTE QSUB SUBNAME LIKE "EMP%" AT NODE 1;
```

Example - using a replication queue map

To promote all Q subscriptions that use the replication queue map qmap1:

```
PROMOTE QSUB REPLQMAP "qmap1";
```

Example - changing to a new replication queue map

To promote all Q subscriptions that use the replication queue map qmap1 so that they use the queue map qmap2 instead:

```
PROMOTE QSUB REPLQMAP "qmap1" USING REPLQMAP "qmap2";
```

PROMOTE REPLQMAP command

Use the PROMOTE REPLQMAP command to promote one or more replication queue maps from one set of control tables to another.

If a single replication queue map is specified, you can also use this command to customize some of the properties of the promoted queue map, including the name of the replication queue map and name of the send queue. The promoted values of properties that cannot be customized are taken from the source replication queue map. If you need to change other properties, you can use the ALTER REPLQMAP command after promoting the replication queue map to change the properties for the new replication queue map.

Syntax

```
►► PROMOTE REPLQMAP NAME replqmap USING new-clause LIKE "predicate" ◄◄
```

new-clause:

```
|-----|  
| REPLQMAP |-----|  
| NAME newqmap | map-options |
```

map-options:

```
|-----|  
| ADMINQ newadminq | SENDQ newsendq | RECVQ newrecvq |
```

Parameters

NAME *replqmap*

Specifies the name of an existing replication queue map to be promoted.

LIKE *"predicate"*

Specifies a list of replication queue map names that match the predicate. All replication queue map names that match the predicate will be promoted.

new-clause

REPLQMAP

Specifies new property values for the promoted replication queue map.

NAME *newqmap*

Specifies a new name for the replication queue map. If you do not specify a new name, then the current replication queue map name is used.

map-options

ADMINQ *newadminq*

Specifies a new name for the administration queue. If you do not specify a new name, then the current administration queue name is used.

SENDQ *newsendq*

Specifies a new name for the send queue. If you do not specify a new name, then the current send queue name is used.

RECVQ *newrecvq*

Specifies a new name for the receive queue. If you do not specify a new name, then the current receive queue name is used.

Example 1

To promote replication queue maps that match the name "SAMPLE_ASN":

```
PROMOTE REPLQMAP LIKE "SAMPLE_ASN%";
```

Example 2

To promote replication queue map REPLQMAP2 and customize several properties of the promoted version of that queue map, so that the new replication queue map name is REPLQMAPNEW2, the new administration queue name is adminqnew2, the new send queue name is sendqnew2, and the new receive queue name is recvqnew2:

```
PROMOTE REPLQMAP NAME REPLQMAP2 USING REPLQMAP NAME REPLQMAPNEW2  
ADMINQ "adminqnew2" SENDQ "sendqnew2" RECVQ "recvqnew2";
```

SET APPLY SCHEMA command

Use the SET APPLY SCHEMA command to set a default Q Apply schema for all task commands.

Syntax

```
▶▶—SET APPLY SCHEMA—TO DEFAULT—▶▶  
                  |  
                  | applieschema |
```

Parameters

TO DEFAULT

Specify to set the Q Apply schema to ASN and to reset any previous SET APPLY SCHEMA commands.

applieschema

Specifies the Q Apply schema name.

Example 1

To reset the default Q Apply schema to ASN:

```
SET APPLY SCHEMA TO DEFAULT
```

Example 2

To set the default Q Apply schema to ASN1:

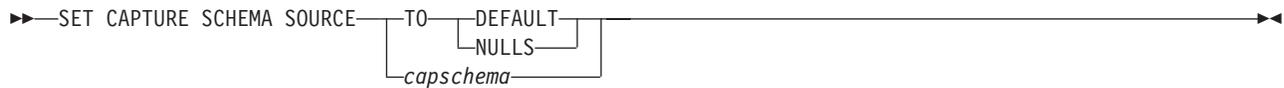
```
SET APPLY SCHEMA ASN1
```

SET CAPTURE SCHEMA command

Use the SET CAPTURE SCHEMA command to set a default schema of the source control tables for all task commands. For Classic sources, you can use only the default Q Capture schema, ASN.

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

Syntax



Parameters

SOURCE

Specifies the Q Capture schema. If you are using a DB2 source, the schema can be any valid DB2 schema name. If you are using a Classic source, you must use the DEFAULT schema.

DEFAULT

Specify to set the Q 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 Q Capture schema name.

Example 1

To reset the default Q Capture schema to ASN:

```
SET CAPTURE SCHEMA SOURCE TO DEFAULT
```

Example 2

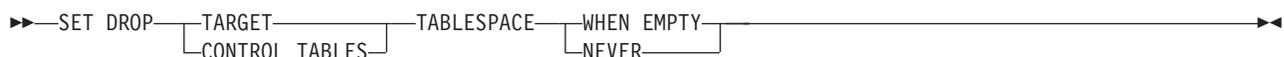
To set the default Q Capture schema to ASN1:

```
SET CAPTURE SCHEMA SOURCE ASN1
```

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



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
```

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 log file name is qrep1msg.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 qmaplog.err for creating replication queue maps:

```
SET LOG "qmaplog.err"
```

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, or the ASNCLP commands needed to promote a replication environment. You cannot use this command with non-relational sources.

Syntax

```
▶▶ SET OUTPUT [CAPTURE SCRIPT "capfname" ] [TARGET SCRIPT "trgfname" ]
▶▶ [PROMOTE SCRIPT "profname" ]
```

Parameters

CAPTURE SCRIPT "*capfname*"

Specifies the output file name for SQL scripts that run at the Q Capture server.

TARGET SCRIPT "*trgfname*"

Specifies the output file name for SQL scripts that run at the Q Apply, or target server.

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.
- The double quotation marks in the command syntax are required.

Example 1

To name the target script output file "target.sql":

```
SET OUTPUT TARGET SCRIPT "target.sql"
```

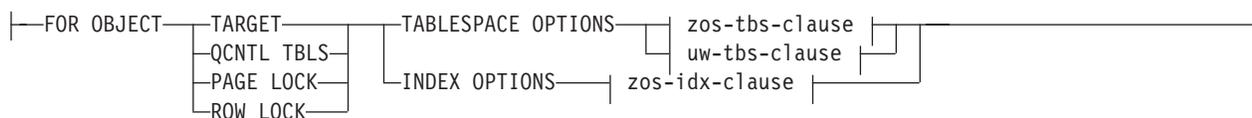
SET PROFILE command

Use the SET PROFILE command to specify custom parameters for table spaces or indexes that are created by the ASNCLP program. After you issue a SET PROFILE command, you can associate a profile with a task command by specifying the profile's name in the task command.

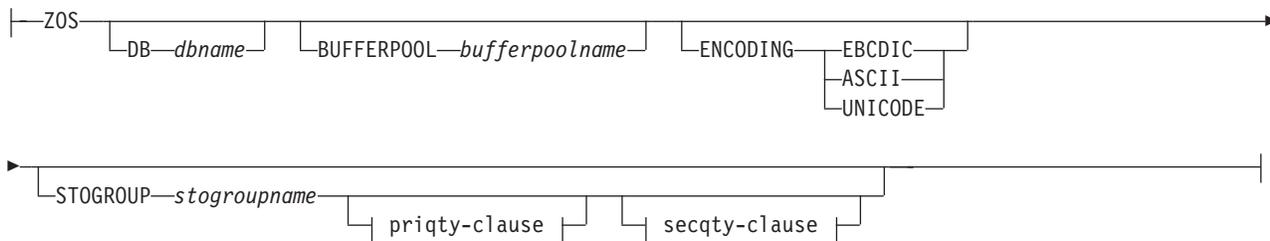
Syntax

```
▶▶ SET PROFILE profilename [prof-clause ]
   [UNDO ]
```

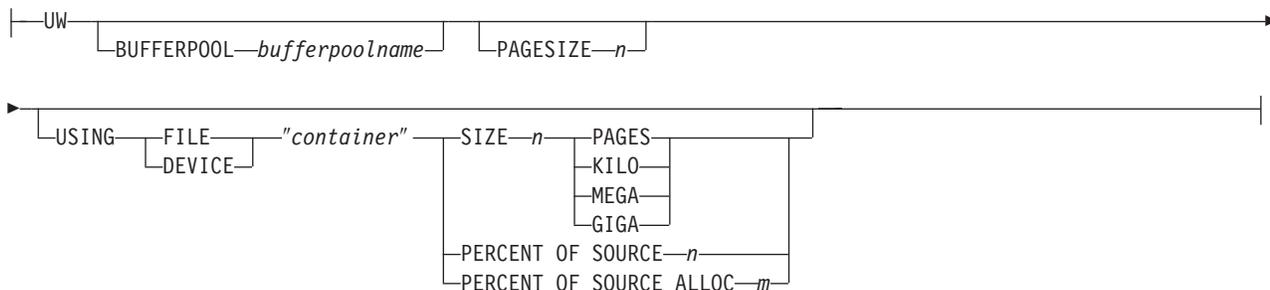
prof-clause:



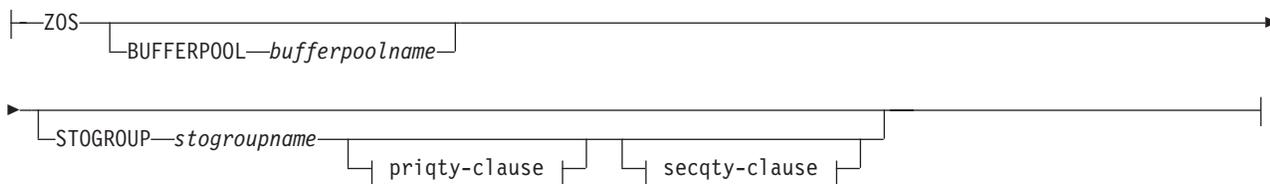
zos-tbs-clause:



uw-tbs-clause:



zos-idx-clause:



priqty-clause:



secqty-clause:



Parameters

PROFILE *profilename*

Specifies the profile name.

UNDO

Specify to undo a specific profile.

FOR OBJECT

Specifies the object for which you are setting table space or index options:

TARGET

Target table

QCNTL TBLS

Q replication control tables

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 table space options.

INDEX OPTIONS

Specify to set index options.

DB *dbname*

Specifies the name of the z/OS database 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

Specifies the minimum primary space allocation for a DB2-managed data set for a table space.

SECQTY

Specifies the minimum secondary space allocation for a DB2-managed data set for a table space.

ABSOLUTE

Specifies an actual value in kilobytes (denoted as *n* or *m* in the syntax diagram) for space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PERCENT OF SOURCE

Specifies the percentage (denoted as *n* or *m* in the syntax diagram) of the source table size for space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PERCENT OF SOURCE ALLOC

The number (denoted as *n* or *m* in the syntax diagram) specifies that the space allocation is at least that percentage of the source table allocation

(not current space usage) of the related source table in z/OS. If it is used in conjunction with the PRIQTY keyword, the number specifies the minimum primary space allocation. If used in conjunction with the SECQTY keyword, the number specifies the minimum secondary space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PAGESIZE *n*

Specifies the page size of the table space.

Restriction: The page size of the table space must match the page size of the buffer pool.

FILE

Specifies the container path string for the file. For example, for Linux or 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 Linux or 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.

SIZE *n*

Specifies the size of the container:

PAGES

Actual number of pages

KILO

Kilobytes

MEGA

Megabytes

GIGA

Gigabytes

Usage notes

- 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 IDXPROFILE that specifies a table space with an 8 kilobytes page size and a 2 gigabyte container for target tables that are created by the ASNCLP program:

```
SET PROFILE IDXPROFILE FOR OBJECT TARGET TABLESPACE OPTIONS UW PAGESIZE 8  
USING FILE "container" SIZE 2 GIGA
```

Example 2

To create a profile TBSPROFILE that sets the index options for tables that follow the page locking mechanism:

```
SET PROFILE TBSPROFILE FOR OBJECT PAGE LOCK INDEX OPTIONS ZOS DB TARGETDB  
STOGROUP MYSTOGROUP PRIQTY PERCENT OF SOURCE 70
```

Example 3

To undo the profile TBSPROFILE:

```
SET PROFILE TBSPROFILE UNDO
```

SET QMANAGER command

Use the SET QMANAGER command to set the WebSphere MQ queue manager. You cannot use this command with non-relational sources.

Syntax

```
▶▶ SET QMANAGER "mgrname" FOR { CAPTURE SCHEMA | APPLY SCHEMA | MULTIDIR servername.schemaname } ▶▶
```

Parameters

"mgrname"

Specifies the name of the WebSphere MQ queue manager.

CAPTURE SCHEMA

Specify to set the queue manager for the Q Capture control tables.

APPLY SCHEMA

Specify to set the queue manager for the Q Apply control tables.

MULTIDIR

Specify to set the queue manager for the bidirectional or peer-to-peer replication server.

servername

Specifies the name of the server (database).

schemaname

Specifies the schema of the control tables.

Example 1

To set the queue manager QM1 for the Q Capture program:

```
SET QMANAGER "QM1" FOR CAPTURE SCHEMA
```

Example 2

To set the queue manager QM2 for the Q Apply program:

```
SET QMANAGER "QM2" FOR APPLY SCHEMA
```

Example 3

To set the queue manager QM3 for a server TESTDB.BLUE that is used in bidirectional or peer-to-peer replication:

```
SET QMANAGER "QM3" FOR MULTIDIR TESTDB.BLUE
```


NO

The ASNCLP program will not generate SQL to create objects that already exist. This is the default.

YES

The ASNCLP program continues to generate SQL statements even if it encounters existing object errors. The following errors are ignored when you specify this option:

Create Control Tables

Another control table already exists under the same schema or table spaces are specified to be created but they already exist.

Create Publication

Another publication with the same name already exists.

Create Publishing Queue Map

Another publishing queue map with the same name already exists.

Create Replication Queue Map

Another replication queue map with the same name already exists.

Create Subscription

Another Q Subscription with the same name already exists, a target table already exists but the option in the CREATE QSUB command is to create the target table, the target table already exists but the option to create the table space was specified, or a unique index with the same name already exists.

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 Q subscriptions or publications 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 might 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 SCRIPT NOW be used.

Figure 2 on page 140 shows these dependencies for Q replication to a relational source. This figure does not apply to non-DB2 sources.

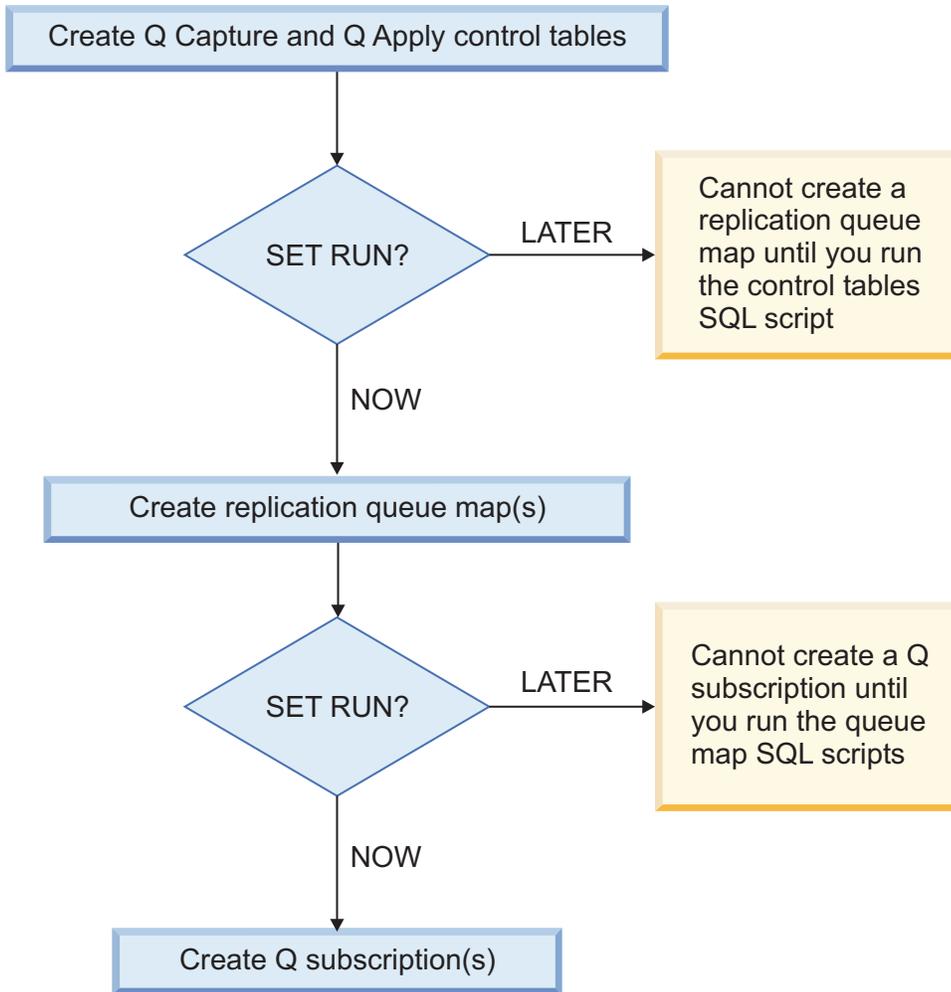


Figure 2. Dependencies between ASNCLP commands for Q replication from a DB2 source. This diagram shows the dependencies between ASNCLP CREATE commands that are used to set up Q replication. It assumes all objects use the default schema of ASN. The dependencies for Q Capture controls tables, publishing queue maps, and publications that are used in event publishing are the same.

Example - Run immediately and stop on errors

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
```

Example - Create SQL script and ignore errors when creating existing objects

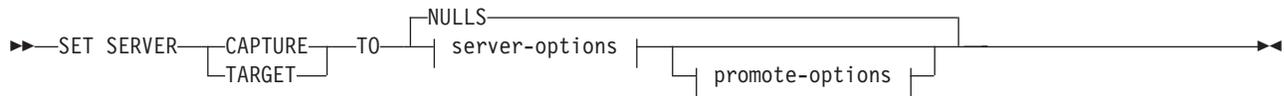
To generate the SQL scripts instead of running them immediately, and to continue generating SQL when creating objects that already exist:

```
SET RUN SCRIPT LATER GENERATE SQL FOR EXISTING YES;
```

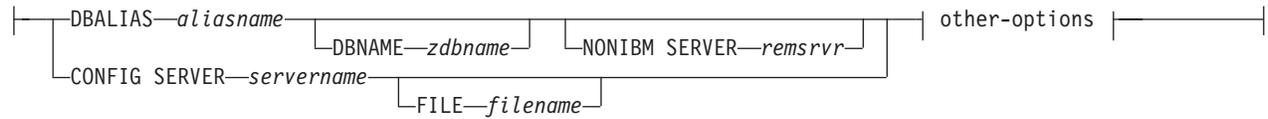
SET SERVER command

Use the SET SERVER command to specify the Q Capture server or Q Apply server (also referred to as a 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.

Syntax



server-options:



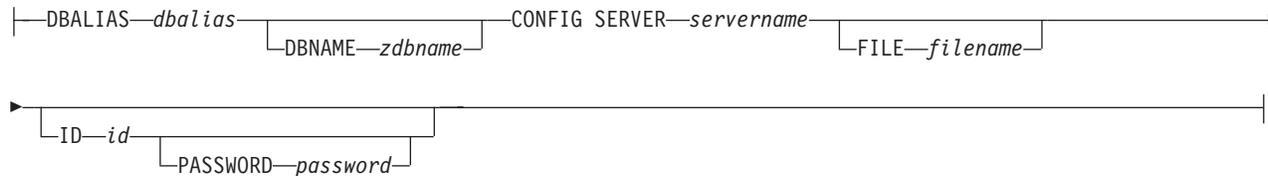
other-options:



promote-options:



promote-srvr-options:



Parameters

CAPTURE

Specify to set the database as a Q Capture or Classic server.

TARGET

Specify to set the database as a Q Apply server.

NULLS

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

server-options:

DBALIAS *aliasname*

Specifies the database alias name.

z/OS

DBNAME *zdbname*

Specifies the database name.

NONIBM SERVER

Federated targets: The remote server name for a federated target. The target can be Informix, Microsoft SQL Server, Oracle, Sybase, or Teradata. This option is only valid for target servers.

CONFIG SERVER *servername*

Classic sources: Specifies the Classic source that the ASNCLP program connects to. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file. You cannot use this parameter if you are using the **TARGET** parameter.

You can also use this parameter to identify DB2 targets.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the **FILE** parameter, the ASNCLP program attempts to use the `asnservers.ini` file in the current directory, if that file exists.

other-options:

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.

promote-options:

PROMOTE TO

Promote the specified server definitions.

SCHEMA *promoteschema*

Specifies the schema under which the server definitions will be promoted. If a schema is not specified, then the schema under which the current server definitions exist is used.

promote-srvr-options:

DBALIAS *dbalias*

Specifies the database that will receive the promoted server definitions. If this clause is not specified and a **PROMOTE** command is included in the input file, then the **PROMOTE** command promotes the definitions to the current server.

z/OS

DBNAME *zdbname*

Specifies the name of the database subsystem that will receive the promoted definitions.

CONFIG SERVER *servername*

Specifies the replication target that the ASNCLP program connects to when promoting definitions. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the **FILE** parameter, the ASNCLP program attempts to use the `asnservers.ini` file in the current directory, if that file exists.

ID *id*

Specifies the database ID where definitions will be promoted to. If not specified, the ASNCLP output script is generated without ID information.

PASSWORD *password*

Specifies the password to use to connect to the database. If not specified, the ASNCLP output script is generated without password information.

Example

To set the Q Capture server to the database SAMPLE:

```
SET SERVER CAPTURE TO DBALIAS SAMPLE;
```

z/OS

Example - z/OS

To set the target server to a z/OS database:

```
SET SERVER TARGET TO DBALIAS PRODUCTION DBNAME PRODUCTIONV9 ID id1 PASSWORD pwd1;
```

This example sets the z/OS database name to PRODUCTIONV9 and specifies the alias PRODUCTION. The user ID and password are explicitly specified because this command sets up a connection to a remote database.

Example - federated targets

To set the target server to an Oracle database ORACLEDB:

```
SET SERVER TARGET TO DBALIAS ORADB NONIBM SERVER ORACLEDB;
```

Example - Classic sources

Given a configuration file called `classic.ini` that contains the following information:

```
[classic1]
Type=CLASSIC
Data source=CACSAMP
Host=9.30.155.156
Port=8019
```

Use the following command to specify server `classic1` as the data server:

```
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE classic.ini ID id1 PASSWORD pwd1;
```

Example - password prompting

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

```
SET SERVER CAPTURE 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. Your text is hidden when you type.

Example - promoting configurations

To set the existing server containing definitions to be promoted and set the new server that will receive these promoted configurations:

```
SET SERVER CAPTURE TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd"
PROMOTE TO DBALIAS SAMPLE1 ID id1 PASSWORD SCHEMA ASN;
```

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

- All output is sent to the console. For readability, save the output to a file.

Example

To turn on the internal trace for the ASNCLP program:

```
SET TRACE ON
```

SHOW SET ENV command

The SHOW SET ENV command displays the environment set during the session. The console displays the environment.

Syntax

```
▶▶ SHOW SET ENV ▶▶
```

Example

To display the environment set during an ASNCLP session:

```
SHOW SET ENV
```

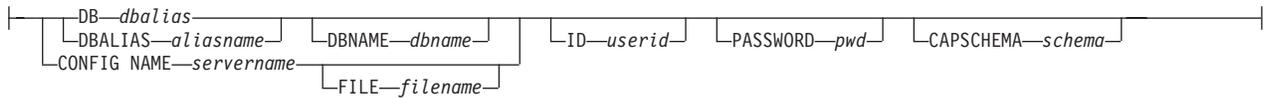
START QSUB command

Use the START QSUB command to signal the Q Capture program or the Classic capture components to start a Q subscription.

Syntax

```
▶▶ START QSUB {SUBNAME subname | FOR SUBNAME LIKE "%text%"} [CAP SERVER OPTIONS classic-opt-clause] ▶▶
```

classic-opt-clause:



Parameters

SUBNAME *subname*

Specifies the name of the Q subscription to start.

FOR SUBNAME LIKE "%text%"

Specify to start Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
START QSUB FOR SUBNAME LIKE "%table%"
```

classic-opt-clause: These parameters work only with Classic sources. If you have already specified these parameters in a previous SET SERVER command, you do not have to specify them again in this command.

DB *dbalias*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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.

CAPSCHEMA *schema*

Specifies the schema of the control tables.

CONFIG NAME *servername*

Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP uses to connect to the Classic data 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 anservers.ini file in the current directory, if that file exists.

Usage notes

The CAP SERVER OPTIONS parameter overrides any settings that you specified in a previous SET command.

Example 1

To start a Q subscription from a Classic source by specifying server information in the START QSUB command:

```
START QSUB SUBNAME sub1 CAP SERVER OPTIONS CONFIG NAME classic1
FILE asnservers.ini ID id1 PASSWORD passwd1;
```

Example 2

To start a Q subscription from a Classic source by specifying server information in a separate SET command:

```
SET SERVER CAPTURE CONFIG SERVER NAME classic1
FILE asnservers.ini ID id1 PASSWORD passwd1;
START QSUB SUBNAME sub1;
```

STOP QSUB command

Use the STOP QSUB command to signal the Q Capture program or the Classic capture components to stop a Q subscription.

Syntax

```
▶▶ STOP QSUB SUBNAME subname
└─ FOR SUBNAME LIKE "%text%" ─┘ └─ CAP SERVER OPTIONS ┘ classic-opt-clause ┘▶▶
```

classic-opt-clause:

```
┌─ DB dbalias ─┘
┌─ DBALIAS aliasname ─┘ ┌─ DBNAME dbname ─┘ ┌─ ID userid ─┘ ┌─ PASSWORD pwd ─┘
└─ CONFIG SERVER servername ─┘ └─ FILE filename ─┘
▶ ┌─ CAPSCHEMA schema ─┘
```

Parameters

SUBNAME *subname*

Specifies the name of the Q subscription to stop.

FOR SUBNAME LIKE "%text%"

Specify to stop Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
STOP QSUB FOR SUBNAME LIKE "%table%"
```

classic-opt-clause: These parameters work only with Classic sources. If you have already specified these parameters in a previous SET SERVER command, you do not have to specify them again in this command.

DB *dbalias*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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.

CAPSCHEMA *schema*

Specifies the schema of the control tables.

CONFIG SERVER *servername*

Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP uses to connect to the Classic data source.

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.

Usage notes

The CAP SERVER OPTIONS parameter overrides any settings that you specified in a previous SET command.

Example

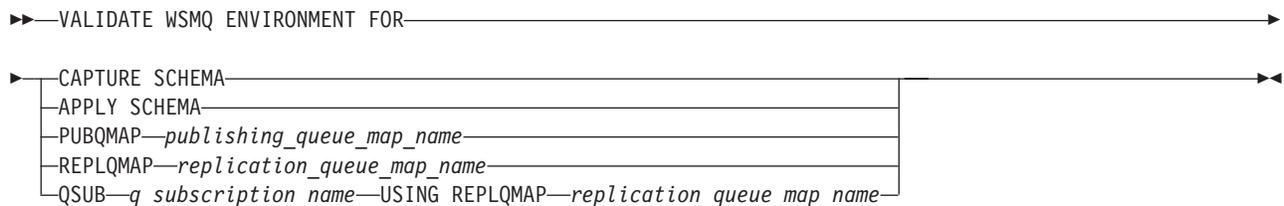
To stop a Q subscription:

```
STOP QSUB SUBNAME EMPLOYEE0001
```

VALIDATE WSMQ ENVIRONMENT FOR command

Use the VALIDATE WSMQ ENVIRONMENT FOR command to verify that the required WebSphere MQ objects exist and have the correct properties for Q replication schemas, queue maps, and Q subscriptions.

Syntax



Parameters

CAPTURE SCHEMA

Specify to validate the queue manager, restart queue, and administration queue that are defined for a Q Capture schema.

APPLY SCHEMA

Specify to validate the queue manager that is defined for a Q Apply schema.

PUBQMAP

Specify to validate the send queue that is specified for a publishing queue map.

REPLQMAP

Specify to validate the send queue, receive queue, and Q Apply administration queue that are specified for a replication queue map.

QSUB

Specify to validate the model queue that is defined to create spill queues for a Q subscription.

Usage notes

Messages that describe the results of the tests are sent to the standard output (stdout).

Example 1

To validate the send queue, receive queue, and Q Apply administration queue that are specified for a replication queue map SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ ENVIRONMENT FOR REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```

Example 2

To validate the model queue that is specified for the Q Subscription EMPLOYEE0001 that uses the replication queue map

SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ ENVIRONMENT FOR QSUB EMPLOYEE0001
USING REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```

VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command

Use the VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command to send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map.

Syntax

```
▶▶—VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP—queue_map_name————▶▶
```

Parameters

queue_map_name

Specifies the name of an existing replication queue map.

Usage notes

The command puts a test message on the send queue and attempts to get the message from the receive queue. It also puts a test message on the Q Apply administration queue and attempts to get the message from the Q Capture administration queue. Messages that describe the results of the tests are sent to the standard output (stdout).

Example

To test the message flow between queues that are part of a replication queue map named SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```


Chapter 5. 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 65
- “Sample ASNCLP scripts for setting up peer-to-peer Q replication (two servers)” on page 70
- “Sample ASNCLP scripts for setting up peer-to-peer Q replication (three servers)” on page 75

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

Description	Command
Change a Q subscription for bidirectional replication	“ALTER QSUB command (bidirectional replication)” on page 154
Change a Q subscription for peer-to-peer replication	“ALTER QSUB command (peer-to-peer replication)” on page 157
Change a replication queue map	ALTER REPLQMAP command
Establish a session for Q replication	ASNCLP SESSION SET TO command
Create the control tables for the Q Capture and Q Apply programs	CREATE CONTROL TABLES FOR command
Create a Q subscription for bidirectional replication	“CREATE QSUB command (bidirectional replication)” on page 167
Create a Q subscription for peer-to-peer replication	“CREATE QSUB command (peer-to-peer replication)” on page 171
Create a replication queue map	CREATE REPLQMAP command
Drop the control tables for the Q Capture and Q Apply programs	DROP CONTROL TABLES ON command
Delete a replication queue map	DROP REPLQMAP command
Delete the subgroup that you set by using the SET SUBGROUP command.	“DROP SUBGROUP command (multidirectional Q replication)” on page 175
Delete a Q subscription for bidirectional replication	“DROP SUBTYPE command (bidirectional replication)” on page 176
Delete a Q subscription for peer-to-peer replication between two servers	“DROP SUBTYPE command (peer-to-peer replication)” on page 176
List Q Apply schemas	LIST APPLY SCHEMA command
List Q Capture schemas	LIST CAPTURE SCHEMA command
Signal that a manual load of the target table is complete	LOAD DONE command
Invoke ASNCLP program scripts used to set up multidirectional replication	“LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication)” on page 180

Table 9. ASNCLP commands for multidirectional Q replication (continued)

Description	Command
Set the Q Apply schema for all task commands	SET APPLY SCHEMA command
Set the Q Capture schema for all task commands	SET CAPTURE SCHEMA command
Connect the servers that are used for bidirectional or peer-to-peer replication.	“SET CONNECTION command (multidirectional Q replication)” on page 188
Specify whether the ASNCLP will enforce matching constraints between the source and target tables.	“SET ENFORCE MATCHING CONSTRAINTS command (multidirectional Q replication)” on page 189
Define the log file for the ASNCLP program	SET LOG command
Specify the Q Capture and Q Apply schema on a server that is used for multidirectional replication	“SET MULTIDIR SCHEMA command (multidirectional Q replication)” on page 190
Define output files that contain SQL scripts for multidirectional replication	“SET OUTPUT command (multidirectional Q replication)” on page 190
Specify custom parameters for database objects to be created implicitly	SET PROFILE command
Set the WebSphere MQ queue manager	SET QMANAGER command
Set a reference table to identify a Q subscription that you want to change or delete.	“SET REFERENCE TABLE command (multidirectional Q replication)” on page 197
Specify the server that contains both Q Capture and Q Apply control tables to use in the ASNCLP session	“SET SERVER command (multidirectional Q replication)” on page 198
Specify the name of the subgroup, a collection of Q subscriptions between servers that are used for multidirectional replication	“SET SUBGROUP command (multidirectional Q replication)” on page 200
Specify the tables that participate in a bidirectional or peer-to-peer configuration	“SET TABLES command (multidirectional Q replication)” on page 200
Enable and disable the trace for the ASNCLP commands	SET TRACE command
Display the environment set during the session	SHOW SET ENV command
Start a Q subscription	START QSUB command
Stop a Q subscription	STOP QSUB command
Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and Q subscriptions.	VALIDATE WSMQ ENVIRONMENT FOR command
Send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map.	VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command

ALTER REPLQMAP command

Use the ALTER REPLQMAP command to customize attributes for an existing replication queue map. This command applies to Q replication and Classic replication.

Syntax

►► ALTER REPLQMAP *qmapname* USING | options | ◀◀

- S** The Q Capture program or the capture components stop when they detect an error on this queue.
- Q** The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues. This value is not supported for Classic replication.

HEARTBEAT INTERVAL *interval*

Specifies the interval (in seconds) between heartbeat messages that are sent by the Q Capture program or Classic capture components to the Q Apply program when there are no transactions to publish.

MAX MESSAGE SIZE *size*

Specifies the maximum size (in kilobytes) of the buffer that is used for sending messages over the send queue. The size of the buffer must not be larger than the maximum message length (MAXMSGL) that is defined for the send queue.

Example 1

The following command alters the SAMPLE_ASN1_TO_TARGETDB_ASN1 replication queue map, sets the threads to 4, and invalidates all of the Q subscriptions that use the send queue for this replication queue map if an error occurs.

```
ASNCLP SESSION SET TO Q REPLICATION;
SET SERVER CAPTURE TO SAMPLE;
SET CAPTURE SCHEMA ASN1;
SET SERVER TARGET TO TARGETDB
SET APPLY SCHEMA ASN1;
ALTER REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1 USING NUM APPLY AGENTS 4
    ERROR ACTION I;
```

Example 2

The following command alters the CLASSIC_ASN_TO_TARGETDB_ASN1 replication queue map, sets the threads to 4, sets the maximum memory limit to 10 megabytes, stops the Classic capture components if an error occurs, sets the heartbeat interval to 4, and sets the maximum buffer size to 5 kilobytes.

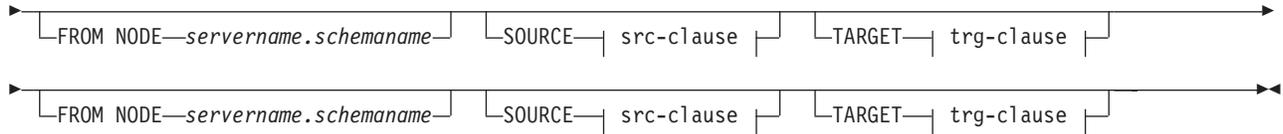
```
ASNCLP SESSION SET TO Q REPLICATION;
SET OUTPUT TARGET SCRIPT "replapp.sql";
SET LOG "qmap.err";
SET SERVER CAPTURE TO CONFIG SERVER server1 FILE "asnservers.ini"
    ID username PASSWORD "passw1rd";
SET SERVER TARGET TO DB TARGETDB;
SET APPLY SCHEMA ASN1;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
ALTER REPLQMAP CLASSIC_ASN_TO_TARGETDB_ASN1 USING NUM APPLY AGENTS 4
    MEMORY LIMIT 10 ERROR ACTION S HEARTBEAT INTERVAL 4 MAX MESSAGE SIZE 5;
```

ALTER QSUB command (bidirectional replication)

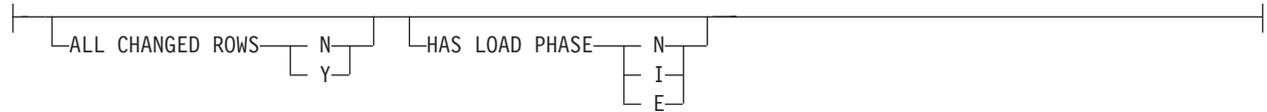
Use the ALTER QSUB command to change the properties of one or both bidirectional Q subscriptions for a single logical table.

Syntax

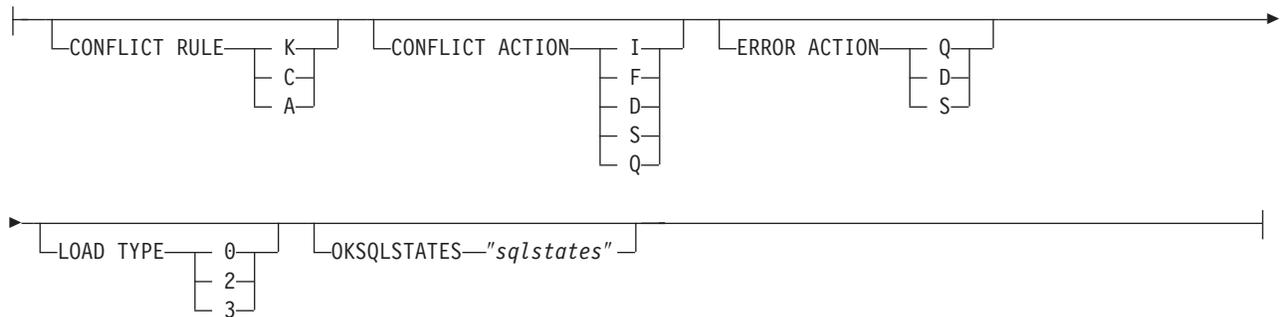
▶▶—ALTER QSUB SUBTYPE B—▶▶



src-clause:



trg-clause:



Parameters

SUBTYPE B

Specifies bidirectional Q subscriptions.

FROM NODE *servername.schemaname*

Identifies one of the two bidirectional Q subscriptions by specifying the server and schema of its source table.

src-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.

trg-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.

(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;
```

ASNCLP SESSION SET TO command

Use the ASNCLP SESSION SET TO command to establish an ASNCLP session for Q replication to either relational or Classic data sources.

Syntax

Parameters

Q REPLICATION

Specify to set the ASNCLP session to Q replication. This ASNCLP session only accepts Q replication syntax. Use this parameter when you are connecting to either relational or Classic sources.

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.
- You can only issue commands that apply to the type of replication that you specify.

Example 1

To set the ASNCLP session to Q replication:

```
ASNCLP SESSION SET TO Q REPLICATION
```

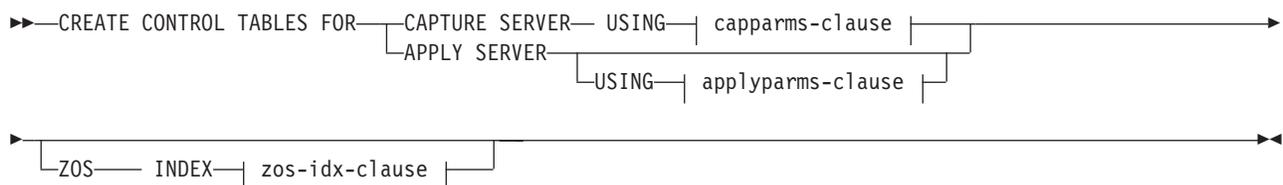
CREATE CONTROL TABLES FOR command

Use the CREATE CONTROL TABLES FOR command to set up Q Capture and Q Apply control tables. For event publishing, Q Apply control tables are not needed.

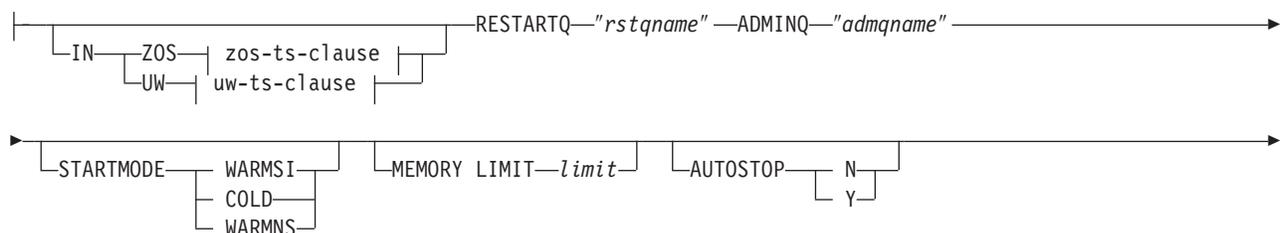
For bidirectional and peer-to-peer replication, run the SET MULTIDIR SCHEMA command before you use this command. The Q Capture and Q Apply programs must use the same schema on each server.

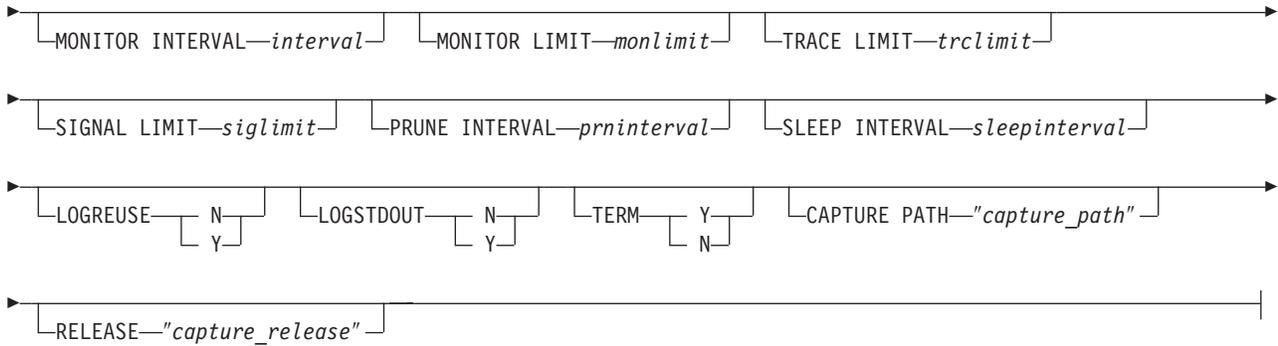
In Classic replication, the control tables for the Classic capture components are creating by using the Classic Data Architect.

Syntax

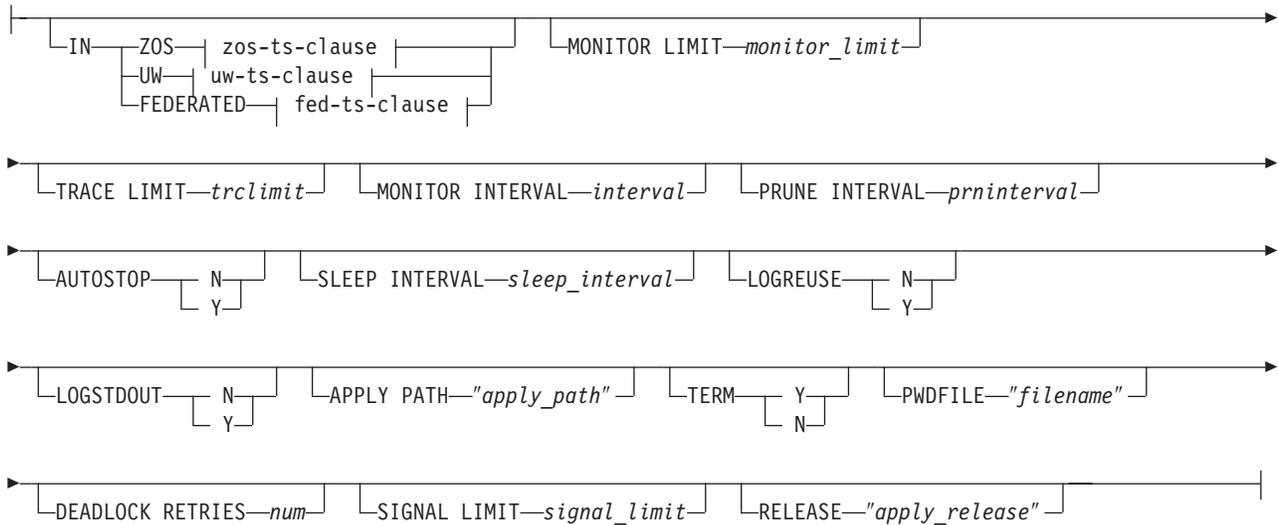


capparms-clause:

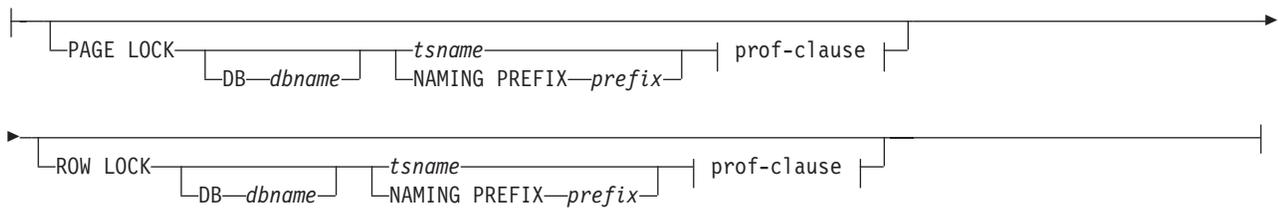




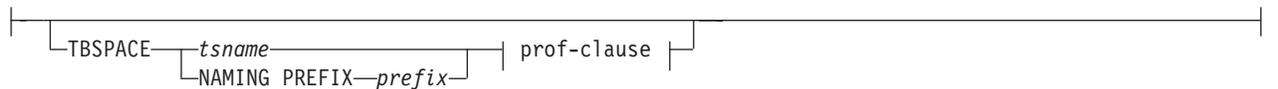
applyparms-clause:



zos-ts-clause:



uw-ts-clause:



fed-ts-clause:



prof-clause:



zos-idx-clause:



Parameters

CAPTURE SERVER

Specify to create Q Capture control tables.

APPLY SERVER

Specify to create Q Apply control tables.

capparms-clause:

ZOS

Specifies a z/OS system on which to create Q Capture control tables.

UW

Specifies a Linux, UNIX, or Windows system on which to create Q Capture control tables.

RESTARTQ "rstqname"

Specifies the restart queue that the Q Capture program uses.

ADMINQ "admqname"

Specifies the administration queue that the Q Capture program uses.

STARTMODE

Specifies what kind of start the Q Capture program will perform.

WARMSI

Specify for the Q Capture program to perform a warm start. If the Q Capture program is starting for the first time, it will perform a cold start.

COLD

Specify for the Q Capture program to perform a cold start.

WARMNS

Specify for the Q Capture program to attempt a warm start if information is available. If the information is not available, the Q Capture program will stop.

MEMORY LIMIT *limit*

Specifies the maximum amount (in MB) of memory that the Q Capture program can use to build transactions.

AUTOSTOP

N The Q Capture or Q Apply program does not stop after it reaches the end of the active log and finds no transactions.

Y The Q Capture or Q Apply program stops after it reaches the end of the active log and finds no transactions.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Capture program inserts rows into the IBMQREP_CAPMON table.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPMON and IBMQREP_CAPQMON tables before it becomes eligible for pruning. All rows in these tables that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

SIGNAL LIMIT *siglimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_SIGNAL table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_CAPMON, IBMQREP_CAPQMON, IBMQREP_CAPTRACE, and IBMQREP_SIGNAL tables are pruned.

SLEEP INTERVAL *sleepinterval*

Specifies the number of milliseconds that the Q Capture program sleeps when it finishes processing the active log and determines that the buffer is empty.

LOGREUSE

- N The Q Capture program appends messages to the log file, even after the Q Capture program restarts.
- Y The Q Capture program reuses the log file by first truncating the current log file and then starting a new log when the Q Capture program restarts.

LOGSTDOUT

- N The Q Capture program only sends messages to the log file.
- Y The Q Capture program sends messages to both the log file and the standard output (stdout).

TERM

- Y The Q Capture program terminates if DB2 is quiesced or stops. This value is the default.
- N The Q Capture program continues running if DB2 is quiesced or stops.

CAPTURE_PATH *"capture_path"*

Specifies the location of the work files that the Q Capture program uses. On z/OS systems, the location can be an MVS data set high-level qualifier with //. The default is NULL.

Linux UNIX Windows RELEASE *"capture_release"*

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks ("). Specifying the release level enables newer replication and publishing function on an older DB2.

appparms-clause:

ZOS

Specifies a z/OS system in which Q Apply control tables are created.

UW

Specifies a Linux, UNIX, or Windows system in which Q Apply control tables are created.

FEDERATED

Specifies a federated target, on which Q Apply control tables are created in an Oracle, Sybase, Informix, Microsoft SQL Server, or Teradata database, and nicknames are created for these control tables in the Q Apply server. Some control tables are created in the Q Apply server.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_APPLYMON table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_APPLYTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Apply program inserts rows into the IBMQREP_APPLYMON table.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_APPLYMON and IBMQREP_APPLYTRACE tables are pruned.

AUTOSTOP

N The Q Apply program does not stop after all queues are emptied once.

Y The Q Apply program stops after all queues are emptied once.

LOGREUSE

N The Q Apply program appends messages to the log file, even after the Q Apply program is restarted.

Y The Q Apply program reuses the log file by first truncating the current log file and then starting a new log when the Q Apply program is restarted.

LOGSTDOUT

N The Q Apply program sends messages only to the log file.

Y The Q Apply program sends messages to the log file and the standard output (stdout).

APPLY PATH "*apply_path*"

Specifies the location of the work files the Q Apply program uses. The default path is the directory where the asnqapp command was run.

TERM

Y The Q Apply program stops if DB2 is quiesced or stops.

N The Q Apply program continues running if DB2 is quiesced or stops.

PWDFILE "*filename*"

Specifies the name of the password file.

DEADLOCK RETRIES *num*

Specifies the number of retries for SQL deadlock errors.

Linux UNIX Windows **RELEASE** "*apply_release*"

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks (""). Specifying the release level enables newer replication and publishing function on an older DB2.

zos-ts-clause:

PAGE LOCK

Specify for replication control tables that require page-level locking.

ROW LOCK

Specify for replication control tables that require row-level locking.

DB *dbname*

Specifies the name of the database that contains the table space where the control tables will be created.

tsname

Specifies the name of the table space for the z/OS control tables.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

uw-ts-clause:

TBSPACE*tsname*

Specifies the name of the table space that is used for the control tables on Linux, UNIX, or Windows.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

fed-ts-clause:

TBSPACE *tsname*

Specifies the name of an existing Oracle table space, Sybase segment, Informix dbspace, or Microsoft SQL Server file group that is used for the control tables. This parameter is not applicable for Teradata targets.

RMT SCHEMA

The remote schema that the Q Apply program uses to create control tables on the non-DB2 database. The default is the remote authorization ID.

CREATE

Specify to create a table space. When this parameter is used without the USING PROFILE keyword, the table space is assumed to exist and the control tables are created in this table space.

USING PROFILE *pname*

Specifies the name of a profile to use to customize the table space attributes.

Example 1

To create Q Apply control tables and to specify a monitor limit of 3 minutes and a trace limit of 9 minutes:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING MONITOR LIMIT 3 TRACE LIMIT 9
```

Example 2

To create Q Capture control tables:

```
CREATE CONTROL TABLES FOR CAPTURE SERVER USING  
RESTARTQ "ASN1.QM1.RESTARTQ" ADMINQ "ASN1.QM1.ADMINQ"
```

Example 3

To create Q Apply control tables for replication to an Oracle target with a remote authorization ID of ORACLE_ID:

```
CREATE CONTROL TABLES FOR APPLY SERVER IN FEDERATED RMT SCHEMA ORACLE_ID
```

Example 4

To create Version 9.7 Q Apply control tables on a DB2 Version 9.1 database:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING RELEASE "9.7"
```

CREATE REPLQMAP command

Use the CREATE REPLQMAP command to create a replication queue map for Q subscriptions.

Syntax

```
▶▶ CREATE REPLQMAP qmapname [DESC "description"] USING ADMINQ "adminqname"  
▶ RECVQ "rcvqname" SENDQ "sendqname" [NUM APPLY AGENTS num] [MAXAGENTS CORRELID num]  
▶ [MEMORY LIMIT limit] [ERROR ACTION S] [HEARTBEAT INTERVAL interval]  
▶ [MAX MESSAGE SIZE size]
```

Parameters

qmapname

Specifies the name of the replication queue map.

DESC "*description*"

Specifies the description of the replication queue map.

ADMINQ "*adminqname*"

Specifies the name of the administration queue at the Q Apply server.

Note: If the Q Capture or the Classic capture components share a single queue manager with the Q Apply program, the programs can share an administration queue.

RECVQ "*rcvqname*"

Specifies the name of the receive queue that is used by the Q Apply program.

SENDQ "*sendqname*"

Specifies the name of the send queue that is used by the Q Capture program (for relational sources) or the capture components.

NUM APPLY AGENTS *num*

Specifies the number of threads that are used for concurrently applying transactions from the specified receive queue.

MAXAGENTS CORRELID *num*

z/OS Specifies that number of threads that are used for concurrently applying transactions from the specified receive queue with the same *correlation ID*. The correlation ID identifies all transactions that were started from the same z/OS job on the Q Capture server.

The value for the **MAXAGENTS CORRELID** parameter cannot be greater than the value for the **NUM APPLY AGENTS** parameter. If

MAXAGENTS_CORRELID value is 1, the transactions will be applied one at a time. If the value is greater than one, for example 4, four agents will apply transactions with the same correlation ID in parallel. If the value is 0, transactions are applied in parallel by using the total number of threads specified by the **NUM APPLY AGENTS** parameter.

MEMORY LIMIT *limit*

Specifies the maximum number of megabytes that are used per receive queue for buffering incoming transactions.

ERROR ACTION

The action that the Q Capture program takes 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.

- S** The Q Capture program or the capture components stop when they detect an error on this queue.
- Q** The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues. This value is not supported for Classic replication.

HEARTBEAT INTERVAL *interval*

Specifies the interval (in seconds) between heartbeat messages that are sent from the Q Capture program or the capture components to the Q Apply program when there are no transactions to publish.

MAX MESSAGE SIZE *size*

Specifies the maximum size (in kilobytes) of the buffer that is used for sending messages over the send queue.

Example 1

To create a replication queue map **SAMPLE_ASN1_TO_TARGETDB_ASN1** from a relational source:

```
CREATE REPLQMAP SAMPLe_ASN1_TO_TARGETDB_ASN1 USING ADMINQ "ASN1.QM1.ADMINQ"
RECVQ "ASN1.QM1_TO_QM2.DATAQ" SENDQ "ASN1.QM1_TO_QM2.DATAQ"
```

Example 2

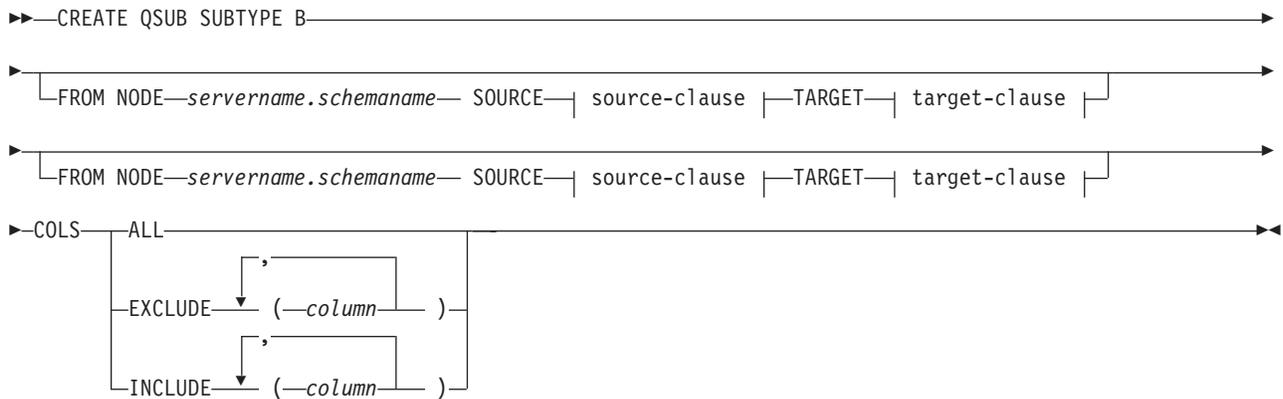
To create a replication queue map **CLASSIC_ASN_TO_TARGETDB_ASN1** from a Classic source:

```
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE classic.ini ID id1 PASSWORD pwd1
SET SERVER TARGET TO DB ASN1
SET RUN SCRIPT NOW STOP ON SQL ERROR ON
CREATE REPLQMAP CLASSIC1_ASN_TO_TARGETDB_ASN1 USING ADMINQ "ASN1.QM1.ADMINQ"
RECVQ "CLASSIC1.QM1_TO_QM2.DATAQ" SENDQ "CLASSIC1.QM1_TO_QM2.DATAQ"
```

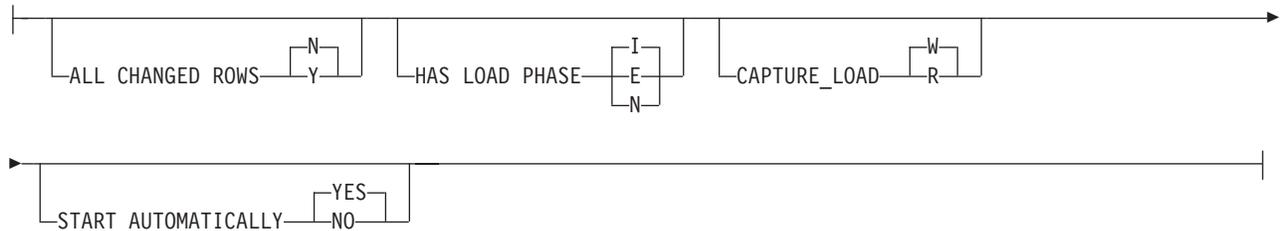
CREATE QSUB command (bidirectional replication)

Use the CREATE QSUB command to create two Q subscriptions for a single logical table that participates in bidirectional replication.

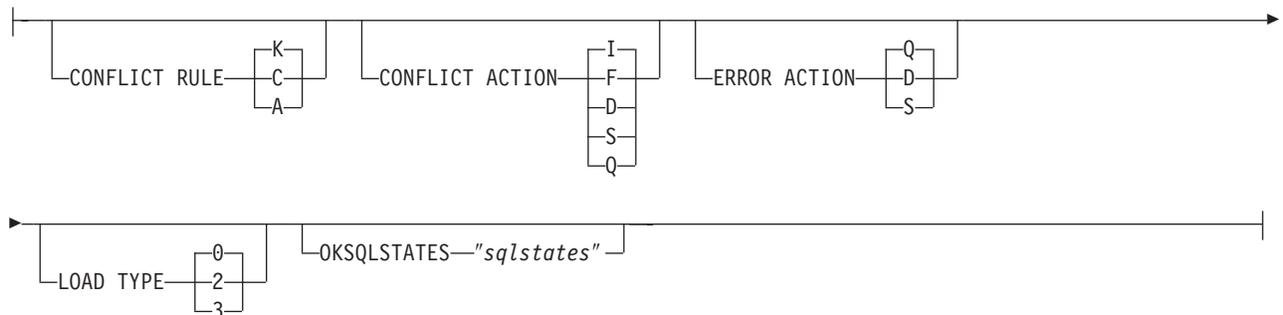
Syntax



source-clause:



target-clause:



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.

COLS

Specifies columns to be selected.

ALL

Select all columns in the subscription. This is the default.

EXCLUDE (*column*)

Exclude the specified columns from the subscription. Columns are excluded from the source table and the target table (if it exists).

INCLUDE (*column*)

Include the specified columns in the 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.

I (default)

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 (by 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.

N No load phase at the target.

CAPTURE_LOAD

Specifies the action that the Q Capture program takes when the recovery log shows that a load operation that uses the DB2 LOAD utility occurred at the source table. This parameter is only valid when the HAS LOAD PHASE option is I.

W (default)

Q Capture issues a warning message after the load completes.

R Q Capture issues a warning message and then stops and starts the Q subscription for the source table, prompting a load of the target table if one is specified for the Q subscription.

START AUTOMATICALLY

Specifies how to start the Q subscription, which is represented by the State column in the IBMQREP_SUBS table. The State column controls whether the subscription is automatically started after starting or reinitializing the Q Capture program (subscription state N), or that the subscription must be started manually by inserting a command in the IBMQREP_SIGNAL table (subscription state I).

YES

The Q subscription is started automatically (subscription state value of N). This is the default.

NO

The Q subscription must be started manually (subscription state value of I).

target-clause:

CONFLICT RULE

- K** Check only key values.
- C** Check changed nonkey values and 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 by using the administration tools.

Excluded columns from either source or target must be defined as nullable or not null with default columns.

Columns defined with data types ROWID and GENERATED ALWAYS are excluded automatically.

Table 10. Required attributes for BEFORE_VALUES and CHANGE_COLS_ONLY depending on the values of CONFLICT_RULE and CONFLICT_ACTION

CONFLICT RULE	CONFLICT ACTION	BEFORE VALUES	CHANGE COLS ONLY
K	I, S, D, or Q	N	Y
K	F	N	N
C	I, S, D, or Q	Y	Y
C	F	Y	N
A	I, S, D, or Q	Y	N

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.

```
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";
```

```
SET TABLES (SAMPLE.RED.REP.EMPLOYEE);
```

```
CREATE QSUB SUBTYPE B
FROM NODE SAMPLE.RED SOURCE HAS LOAD PHASE I
TARGET CONFLICT RULE C CONFLICT ACTION F
FROM NODE SAMPLE2.BLUE SOURCE HAS LOAD PHASE I
TARGET CONFLICT RULE A CONFLICT ACTION I
```

Example: Subsetting columns - include

The following command creates a bidirectional Q subscription that includes only columns c1, c2, c3, and c4:

```
CREATE QSUB SUBTYPE B COLS INCLUDE (C1,C2,C3,C4)
```

If the target table is new, the table is created with the specified columns. If the target table exists, then the specified columns are included in the table. The column names in the source table and target table must be the same.

Example: Subsetting columns - exclude

The following command creates a bidirectional Q subscription that excludes columns C1, C2, and C3:

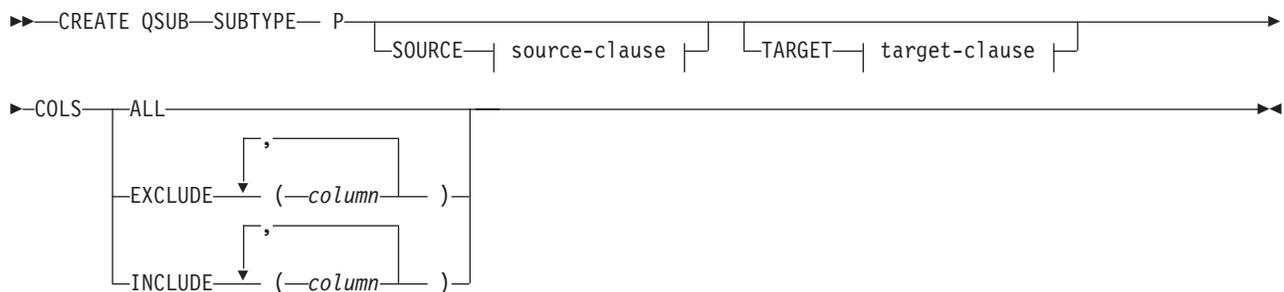
```
CREATE QSUB SUBTYPE B SOURCE HAS LOAD PHASE I COLS EXCLUDE (C1,C2,C3)
```

The specified columns are excluded from the source table and target table (if the target exists). The column names in the source table and target table must be the same. Excluded columns must be nullable, or if defined as NOT NULL then they must have a default value.

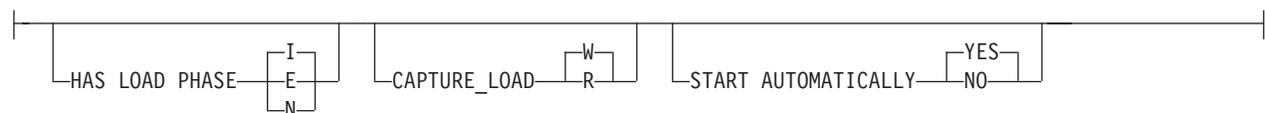
CREATE QSUB command (peer-to-peer replication)

Use the CREATE QSUB command to create a set of Q subscriptions for a single logical table that participates in peer-to-peer replication.

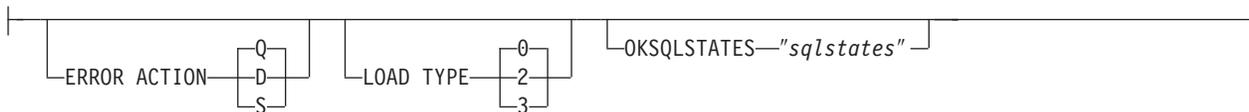
Syntax



source-clause:



target-clause:



Parameters

SUBTYPE P

Specifies Q subscriptions for peer-to-peer replication.

COLS

Specifies columns to be selected.

ALL

Select all columns in the subscription. This is the default.

EXCLUDE (*column*)

Exclude the specified columns from the subscription. Columns are excluded from the source table and the target table (if it exists).

INCLUDE (*column*)

Include the specified columns in the subscription.

source-clause:

HAS LOAD PHASE

Specifies whether the tables that are specified in the Q subscriptions will be loaded with data from one of the peer copies of the table.

I (default)

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.

N No load phase.

CAPTURE_LOAD

For peer-to-peer replication with two servers only: Specifies the action that the Q Capture program takes when the recovery log shows that a load operation that uses the DB2 LOAD utility occurred at the source table.

W (default)

Q Capture issues a warning message after the load completes.

R Q Capture issues a warning message and then stops and starts the Q subscription for the source table, prompting a load of the target table if a load is specified for the Q subscription.

START AUTOMATICALLY

Specifies how to start the Q subscription, which is represented by the State column in the IBMQREP_SUBS table. The State column controls whether the subscription is automatically started after starting or reinitializing the Q Capture program (subscription state N), or that the subscription must be started manually by inserting a command in the IBMQREP_SIGNAL table (subscription state I).

YES

The Q subscription is started automatically (subscription state value of N). This is the default.

NO

The Q subscription must be started manually (subscription state value of I).

target-clause:

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

Conflict Rule	Conflict Action	Before Values	Change Cols Only
V	F	N	N

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;
```


Usage notes

This command is used in conjunction with the SET SERVER command to indicate the location of the control tables.

Example

To drop the Q Capture control tables:

```
SET SERVER TARGET TO QAPPDB;  
DROP CONTROL TABLES ON APPLY SERVER
```

DROP REPLQMAP command

Use the DROP REPLQMAP command to delete existing replication queue maps.

Restriction: Before you use the DROP REPLQMAP command, delete all Q subscriptions that use the replication queue map.

Syntax

```
►►—DROP REPLQMAP—qmapname—◄◄
```

Parameters

qmapname

Specifies the name of the replication queue map to delete.

Example

To delete the SAMPLE_ASN1_TO_TARGETDB_ASN1 replication queue map:

```
DROP REPLQMAP SAMPLE_ASN1_TO_TARGETDB_ASN1
```

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;
```

LIST APPLY SCHEMA command

You can use the LIST APPLY SCHEMA command to list the Q Apply schemas for a specified server.

Syntax

►► LIST APPLY SCHEMA ◀◀

┌──SERVER──┐ dbparms ┌──┐

dbparms-clause:

┌──DBALIAS──*aliasname*──┐ ┌──DBNAME──*dbname*──┐ ┌──ID──*userid*──┐ ┌──PASSWORD──*pwd*──┐
└──CONFIG SERVER──*servername*──┘ └──FILE──*filename*──┘

Parameters

dbparms-clause:

SERVER

Specifies the server that contains the schemas to be listed.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

Example

To list the Q Capture schema on server SAMPLE:

```
LIST CAPTURE SCHEMA SERVER DBALIAS SAMPLE ID id1 PASSWORD "passwd!";
```

LIST CAPTURE SCHEMA command

You can use the LIST CAPTURE SCHEMA command to list the Q Capture schemas for a specified server.

Syntax

```
▶▶ LIST CAPTURE SCHEMA [SERVER—server] [dbparms] ▶▶
```

dbparms-clause:

```
[DBALIAS—aliasname] [DBNAME—dbname] [ID—userid] [PASSWORD—pwd]  
[CONFIG SERVER—servername] [FILE—filename]
```

Parameters

dbparms-clause:

SERVER

Specifies the server that contains the schemas to be listed.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

Example

To list the Q Capture schema on server SAMPLE:

```
LIST CAPTURE SCHEMA SERVER DBALIAS SAMPLE ID id1 PASSWORD "passwd!";
```

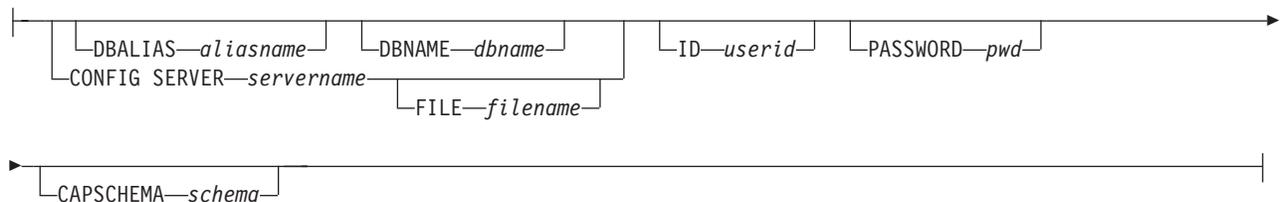
LOAD DONE command

Use the LOAD DONE command to inform the Q Capture program or the Classic capture components that the target table is loaded. Issue the LOAD DONE command only if you are doing a manual load. If the Q Apply program is doing the load, this signal is not necessary.

Syntax



classic-opt-clause:



Parameters

SUBNAME *subname*

Specifies the name of the Q subscription for the LOADDONE signal.

FOR SUBNAME LIKE "*%text%*"

Specify to signal that the load is done for Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
LOAD DONE QSUB FOR SUBNAME LIKE "%table%"
```

CAP SERVER OPTIONS

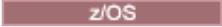
Specifies additional parameters when you issue the LOAD DONE command in immediate execution mode.

classic-opt-clause: These parameters only work with Classic sources.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

 Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

ID *userid*

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

PASSWORD *pwd*

Specifies the password to use to connect to the source database.

CAPSCHEMA *schema*

Specifies the schema of the control tables of the Classic source.

CONFIG SERVER *servername*

Specifies which server configuration settings from the Classic replication configuration file that the ASNCLP should use to connect to the Classic data source.

FILE *filename*

Specifies the Classic replication server that the ASNCLP program connects to. The server name must match the name that is entered in the Classic 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.

Example

To signal the Q Capture program or the capture components that the target table for the Q subscription EMPLOYEE0001 is loaded:

```
LOAD DONE QSUB SUBNAME EMPLOYEE0001
```

LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication)

Use the LOAD MULTIDIR REPL SCRIPT command to invoke ASNCLP program scripts used to set up peer-to-peer and bidirectional replication.

Syntax

```
▶▶—LOAD MULTIDIR REPL SCRIPT—"filelocation/filename"—▶▶  
                                  └─"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 REPL SCRIPT call.
- Several LOAD MULTIDIR REPL 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 REPL SCRIPT "3nodes\3Node0.in";  
LOAD MULTIDIR REPL SCRIPT "3nodes\3Node1.in";  
LOAD MULTIDIR REPL SCRIPT "3nodes\3Node2.in";  
LOAD MULTIDIR REPL 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 "testdb2".YELLOW replqmap "BLUEtoYELLOW";  
set connection SOURCE "testdb1".RED TARGET "testdb".BLUE replqmap "REDtoBLUE";  
set connection SOURCE "testdb1".RED TARGET "testdb2".YELLOW replqmap "REDtoYELLOW";  
set connection SOURCE "testdb2".YELLOW TARGET "testdb".BLUE replqmap "YELLOWtoBLUE";  
set connection SOURCE "testdb2".YELLOW TARGET "testdb1".RED 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;

```

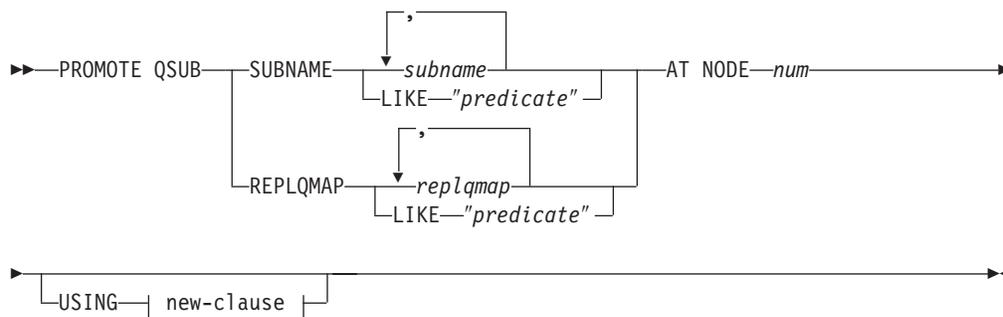
This bidirectional or peer-to-peer script creates a subgroup "3Node0". All of the information required to generate the subgroup's Q subscriptions is located in this one input file.

PROMOTE QSUB command

Use the PROMOTE QSUB command to build an ASNCLP script with statements that you can use to create Q subscriptions on another set of servers. Promoting is useful for copying Q subscriptions from test systems to production systems or migrating Q subscriptions from one server to another.

You can also use this command to customize some of the properties of the promoted Q subscription, including the name of the Q Capture and Q Apply schemas and the replication queue map that is used. The promoted values of properties that cannot be customized are taken from the source Q subscription. If you need to change other properties, you can use the ALTER QSUB command after promoting the Q subscription to change the properties for the new Q subscription.

Syntax



new-clause::



Parameters

SUBNAME *subname*

Specifies one or more Q subscription names to promote. Separate multiple Q subscription names with commas.

LIKE *"predicate"*

Specifies a list of Q subscription names to promote that match the predicate.

REPLQMAP *replqmap*

Specifies one or more replication queue maps. Separate multiple map names with commas. All Q subscriptions that use the specified map or maps are promoted.

LIKE "predicate"

Specifies a list replication queue maps that match the predicate. All Q subscriptions that use the matching maps are promoted.

AT NODE *num*

Specifies the node number of the source configuration to be promoted. Default value is 1 for NODE 1. This option is ignored when you promote unidirectional Q subscriptions.

A "node" is a paired Q Capture/Q Apply schema at a server that is participating in bidirectional or peer-to-peer replication. For example you could have three physical machines that were involved in peer-to-peer replication, each with a DB2 database. Within each database are one or more Q Capture/Q Apply programs and their control tables that are identified by a schema. The paired schema represents a "node" in a three-way peer-to-peer configuration

new-clause:

USING SOURCE SCHEMA *schema*

Specifies the source table schema.

USING TARGET SCHEMA *schema*

Specifies the target table schema. If the schema is not specified, the promoted definition uses the schema of the current target table.

USING REPLQMAP *newqmap*

Specifies the name of a new replication queue map that you want to use for the promoted Q subscriptions.

Example - matching a predicate

To promote all bidirectional Q subscriptions that match the predicate EMP at NODE 1:

```
PROMOTE QSUB SUBNAME LIKE "EMP%" AT NODE 1;
```

Example - using a replication queue map

To promote all Q subscriptions that use the replication queue map qmap1:

```
PROMOTE QSUB REPLQMAP "qmap1";
```

Example - changing to a new replication queue map

To promote all Q subscriptions that use the replication queue map qmap1 so that they use the queue map qmap2 instead:

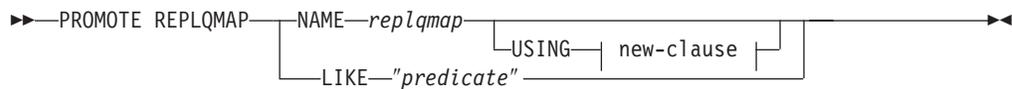
```
PROMOTE QSUB REPLQMAP "qmap1" USING REPLQMAP "qmap2";
```

PROMOTE REPLQMAP command

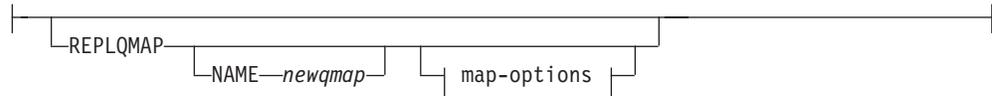
Use the PROMOTE REPLQMAP command to promote one or more replication queue maps from one set of control tables to another.

If a single replication queue map is specified, you can also use this command to customize some of the properties of the promoted queue map, including the name of the replication queue map and name of the send queue. The promoted values of properties that cannot be customized are taken from the source replication queue map. If you need to change other properties, you can use the ALTER REPLQMAP command after promoting the replication queue map to change the properties for the new replication queue map.

Syntax



new-clause:



map-options:



Parameters

NAME *replqmap*

Specifies the name of an existing replication queue map to be promoted.

LIKE *"predicate"*

Specifies a list of replication queue map names that match the predicate. All replication queue map names that match the predicate will be promoted.

new-clause

REPLQMAP

Specifies new property values for the promoted replication queue map.

NAME *newqmap*

Specifies a new name for the replication queue map. If you do not specify a new name, then the current replication queue map name is used.

map-options

ADMINQ *newadminq*

Specifies a new name for the administration queue. If you do not specify a new name, then the current administration queue name is used.

SENDQ *newsendq*

Specifies a new name for the send queue. If you do not specify a new name, then the current send queue name is used.

RECVQ *newrecvq*

Specifies a new name for the receive queue. If you do not specify a new name, then the current receive queue name is used.

Example 1

To promote replication queue maps that match the name "SAMPLE_ASN":

```
PROMOTE REPLQMAP LIKE "SAMPLE_ASN%";
```

Example 2

To promote replication queue map REPLQMAP2 and customize several properties of the promoted version of that queue map, so that the new replication queue map name is REPLQMAPNEW2, the new administration queue name is adminqnew2, the new send queue name is sendqnew2, and the new receive queue name is recvqnew2:

```
PROMOTE REPLQMAP NAME REPLQMAP2 USING REPLQMAP NAME REPLQMAPNEW2  
ADMINQ "adminqnew2" SENDQ "sendqnew2" RECVQ "recvqnew2";
```

SET APPLY SCHEMA command

Use the SET APPLY SCHEMA command to set a default Q Apply schema for all task commands.

Syntax

```
►► SET APPLY SCHEMA [TO DEFAULT] applieschema ◀◀
```

Parameters

TO DEFAULT

Specify to set the Q Apply schema to ASN and to reset any previous SET APPLY SCHEMA commands.

applieschema

Specifies the Q Apply schema name.

Example 1

To reset the default Q Apply schema to ASN:

```
SET APPLY SCHEMA TO DEFAULT
```

Example 2

To set the default Q Apply schema to ASN1:

```
SET APPLY SCHEMA ASN1
```

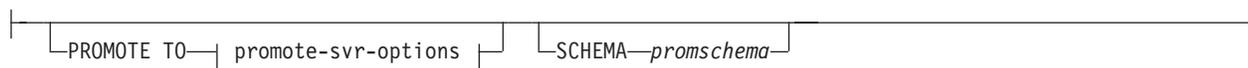
SET BIDI NODE command

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.

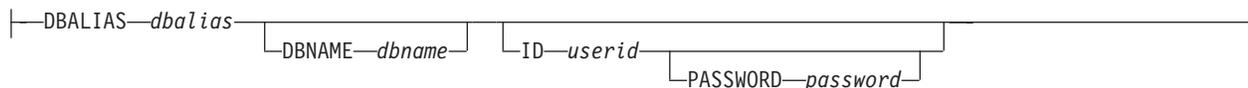
Syntax

```
►► SET [BIDI | BIDIRECTIONAL] NODE number SERVER DBALIAS dbalias [DBNAME dbname] ◀◀  
◀ [ID userid] [PASSWORD pwd] [SCHEMA schema] [promote-options] ◀◀
```

promote-options:



promote-srvr-options:



Parameters

NODE *number*

Specifies server 1 or 2 of the bidirectional configuration. A server 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.

z/OS

DBNAME *dbname*

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.

promote-options

PROMOTE TO *dbalias*

Specifies the destination database alias name to receive the promoted configuration.

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.

promote-srvr-options

DBALIAS *aliasname*

Specifies the destination database alias name.

z/OS

DBNAME *dbname*

Specifies the destination database or subsystem name.

ID *userid*

Specifies the user ID of the database where you want to promote the configurations. This connection information is used in the generated ASNCLP scripts.

PASSWORD *pwd*

Specifies the password of the database where you want to promote the configurations. The user ID and password are used in the generated ASNCLP scripts that you later run to create the new configurations at the destination server.

Example

To specify the servers to promote configurations from and the corresponding destination servers:

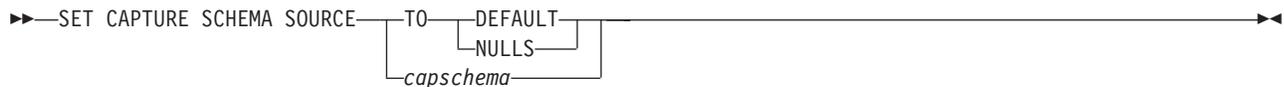
```
SET BIDI NODE 1 SERVER DBALIAS TEST01 ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS PRODUCTION01 ID id1 PASSWORD "pw1d" SCHEMA ASN;
```

```
SET BIDI NODE 2 SERVER DBALIAS TEST02 ID id1 PASSWORD "p1wd" SCHEMA ASN
PROMOTE TO DBALIAS TEST011 ID id1 PASSWORD "pw1d" SCHEMA ASN;
```

SET CAPTURE SCHEMA command

Use the SET CAPTURE SCHEMA command to set a default schema of the source control tables for all task commands. For Classic sources, you can use only the default Q Capture schema, ASN.

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

Syntax**Parameters****SOURCE**

Specifies the Q Capture schema. If you are using a DB2 source, the schema can be any valid DB2 schema name. If you are using a Classic source, you must use the DEFAULT schema.

DEFAULT

Specify to set the Q 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 Q Capture schema name.

Example 1

To reset the default Q Capture schema to ASN:

```
SET CAPTURE SCHEMA SOURCE TO DEFAULT
```

Example 2

To set the default Q Capture schema to ASN1:

```
SET CAPTURE SCHEMA SOURCE ASN1
```

SET CONNECTION command (multidirectional Q replication)

Use the SET CONNECTION command to connect the two servers that are used for bidirectional or peer-to-peer replication.

Syntax

```
►► 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 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—"logfile"—▶▶
```

Parameters

"logfilename"

Specifies the output log file name. The default log file name is qrep1msg.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 qmaplog.err for creating replication queue maps:

```
SET LOG "qmaplog.err"
```

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┐ ┌PROMOTE SCRIPT—"profname"┐  
▶▶────────────────────────────────────────────────────────────────────────────────▶▶
```

Parameters

MULTIDIR

Specify to name the output files after the databases that the SQL scripts run on.

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 `qrep1_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 dbalias DBNAME dbname
    PEERTOPEER
▶ ID userid PASSWORD pwd SCHEMA schema promote-options
```

promote-options:

```
| PROMOTE TO promote-svr-options SCHEMA promschema |
```

promote-svr-options:

```
| DBALIAS dbalias DBNAME dbname ID userid PASSWORD password |
```

Parameters

NODE *number*

Specify with a digit from 1 to 6 a server in the peer-to-peer configuration that

defines the overall peer-to-peer context to be promoted. A server 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 servers 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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 *dbalias*

Specifies the destination database alias name to receive the promoted configuration.

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.

promote-srvr-options

DBALIAS *aliasname*

Specifies the destination database alias name.

z/OS **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 servers to promote configurations from and the corresponding destination servers:

```
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 PROFILE command

Use the SET PROFILE command to specify custom parameters for table spaces or indexes that are created by the ASNCLP program. After you issue a SET PROFILE 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 ] [ UNDO ]
```

prof-clause:

```
| FOR OBJECT [ TARGET [ QCNTL TBLS [ PAGE LOCK [ ROW LOCK ] ] ] ] [ TABLESPACE OPTIONS [ zos-tbs-clause ] ] [ INDEX OPTIONS [ zos-idx-clause ] ] [ uw-tbs-clause ]
```

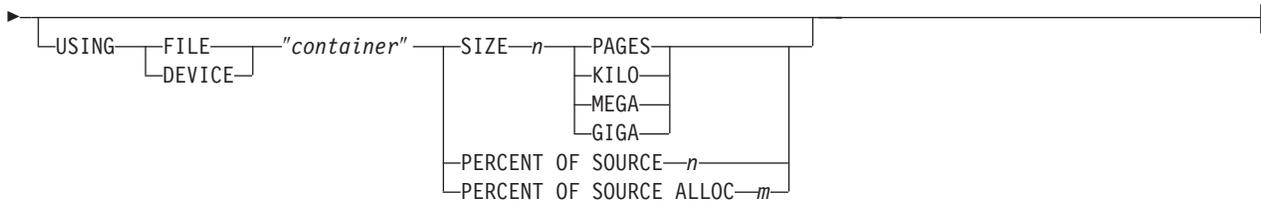
zos-tbs-clause:

```
| ZOS [ DB dbname ] [ BUFFERPOOL bufferpoolname ] [ ENCODING [ EBCDIC ] [ ASCII ] [ UNICODE ] ]
```

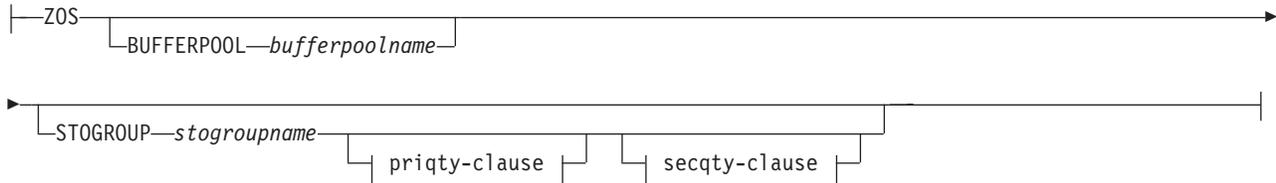
```
▶ [ STOGROUP stogroupname ] [ priqty-clause ] [ secqty-clause ]
```

uw-tbs-clause:

```
| UW [ BUFFERPOOL bufferpoolname ] [ PAGESIZE n ]
```



zos-idx-clause:



priqty-clause:



secqty-clause:



Parameters

PROFILE *profilename*
 Specifies the profile name.

UNDO
 Specify to undo a specific profile.

FOR OBJECT
 Specifies the object for which you are setting table space or index options:

TARGET
 Target table

QCNTL TBLS
 Q replication control tables

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 table space options.

INDEX OPTIONS

Specify to set index options.

DB *dbname*

Specifies the name of the z/OS database 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

Specifies the minimum primary space allocation for a DB2-managed data set for a table space.

SECQTY

Specifies the minimum secondary space allocation for a DB2-managed data set for a table space.

ABSOLUTE

Specifies an actual value in kilobytes (denoted as *n* or *m* in the syntax diagram) for space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PERCENT OF SOURCE

Specifies the percentage (denoted as *n* or *m* in the syntax diagram) of the source table size for space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PERCENT OF SOURCE ALLOC

The number (denoted as *n* or *m* in the syntax diagram) specifies that the space allocation is at least that percentage of the source table allocation (not current space usage) of the related source table in z/OS. If it is used in conjunction with the PRIQTY keyword, the number specifies the minimum primary space allocation. If used in conjunction with the SECQTY keyword, the number specifies the minimum secondary space allocation. See the CREATE TABLESPACE command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PAGESIZE *n*

Specifies the page size of the table space.

Restriction: The page size of the table space must match the page size of the buffer pool.

FILE

Specifies the container path string for the file. For example, for Linux or 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 Linux or 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.

SIZE *n*
Specifies the size of the container:

PAGES
Actual number of pages

KILO
Kilobytes

MEGA
Megabytes

GIGA
Gigabytes

Usage notes

- 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 `IDXPROFILE` that specifies a table space with an 8 kilobytes page size and a 2 gigabyte container for target tables that are created by the ASNCLP program:

```
SET PROFILE IDXPROFILE FOR OBJECT TARGET TABLESPACE OPTIONS UW PAGESIZE 8  
USING FILE "container" SIZE 2 GIGA
```

Example 2

To create a profile `TBSPROFILE` that sets the index options for tables that follow the page locking mechanism:

```
SET PROFILE TBSPROFILE FOR OBJECT PAGE LOCK INDEX OPTIONS ZOS DB TARGETDB  
STOGROUP MYSTOGROUP PRIQTY PERCENT OF SOURCE 70
```

Example 3

To undo the profile `TBSPROFILE`:

```
SET PROFILE TBSPROFILE UNDO
```

SET QMANAGER command

Use the `SET QMANAGER` command to set the WebSphere MQ queue manager. You cannot use this command with non-relational sources.

Syntax

```
▶▶ SET QMANAGER—"mgrname"—FOR—

|                                          |
|------------------------------------------|
| CAPTURE SCHEMA—                          |
| APPLY SCHEMA—                            |
| MULTIDIR— <i>servername.schemaname</i> — |

▶▶
```

Parameters

"mgrname"
Specifies the name of the WebSphere MQ queue manager.

CAPTURE SCHEMA

Specify to set the queue manager for the Q Capture control tables.

APPLY SCHEMA

Specify to set the queue manager for the Q Apply control tables.

MULTIDIR

Specify to set the queue manager for the bidirectional or peer-to-peer replication server.

servername

Specifies the name of the server (database).

schemaname

Specifies the schema of the control tables.

Example 1

To set the queue manager QM1 for the Q Capture program:

```
SET QMANAGER "QM1" FOR CAPTURE SCHEMA
```

Example 2

To set the queue manager QM2 for the Q Apply program:

```
SET QMANAGER "QM2" FOR APPLY SCHEMA
```

Example 3

To set the queue manager QM3 for a server TESTDB.BLUE that is used in bidirectional or peer-to-peer replication:

```
SET QMANAGER "QM3" FOR MULTIDIR TESTDB.BLUE
```

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";

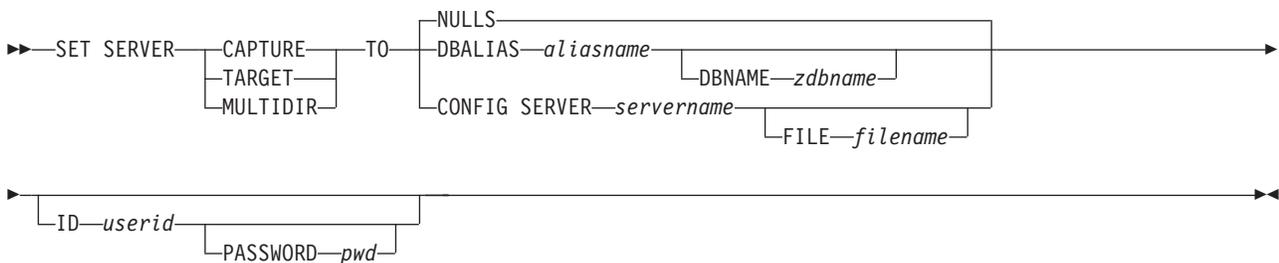
SET REFERENCE TABLE USING SCHEMA "SAMPLE".RED USES TABLE RED.EMPLOYEE;

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



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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

CONFIG SERVER *servername*

DB2 for z/OS and DB2 for Linux, UNIX, and Windows sources only:

Specifies that you are using a file to provide connection information for the server. This keyword is required when the ASNCLP program is running on UNIX System Services (USS) for z/OS. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the FILE parameter, the ASNCLP program attempts to use the asnservers.ini file in the current directory, if that file exists.

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

▶▶—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.RCTEST2.TABLE2, "testdb1".RED.RCTEST3.XYZ,
"testdb2".YELLOW.RCBLUE.AllTypes0);
CREATE QSUB SUBTYPE P;
```

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

- All output is sent to the console. For readability, save the output to a file.

Example

To turn on the internal trace for the ASNCLP program:

```
SET TRACE ON
```

SHOW SET ENV command

The SHOW SET ENV command displays the environment set during the session. The console displays the environment.

Syntax

```
▶▶—SHOW SET ENV—▶▶
```

Example

To display the environment set during an ASNCLP session:

```
SHOW SET ENV
```

START QSUB command

Use the START QSUB command to signal the Q Capture program or the Classic capture components to start a Q subscription.

Syntax

```
▶▶—START— QSUB—SUBNAME—subname—▶▶  
└─FOR SUBNAME LIKE—"%text%"┘ └─CAP SERVER OPTIONS—| classic-opt-clause ┘
```

classic-opt-clause:

```
|└─DB—dbalias—  
└─DBALIAS—aliasname┘ └─DBNAME—dbname┘ └─ID—userid┘ └─PASSWORD—pwd┘ └─CAPSCHEMA—schema┘  
└─CONFIG NAME—servername┘ └─FILE—filename┘
```

Parameters

SUBNAME *subname*

Specifies the name of the Q subscription to start.

FOR SUBNAME LIKE "%text%"

Specify to start Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
START QSUB FOR SUBNAME LIKE "%table%"
```

classic-opt-clause: These parameters work only with Classic sources. If you have already specified these parameters in a previous SET SERVER command, you do not have to specify them again in this command.

DB *dbalias*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

Example

To stop a Q subscription:

```
STOP QSUB SUBNAME EMPLOYEE0001
```

VALIDATE WSMQ ENVIRONMENT FOR command

Use the VALIDATE WSMQ ENVIRONMENT FOR command to verify that the required WebSphere MQ objects exist and have the correct properties for Q replication schemas, queue maps, and Q subscriptions.

Syntax

```
▶▶—VALIDATE WSMQ ENVIRONMENT FOR—————▶▶
▶—CAPTURE SCHEMA—————▶▶
|—APPLY SCHEMA—————▶▶
|—PUBQMAP—publishing_queue_map_name————▶▶
|—REPLQMAP—replication_queue_map_name————▶▶
|—QSUB—q_subscription_name—USING REPLQMAP—replication_queue_map_name—▶▶
```

Parameters

CAPTURE SCHEMA

Specify to validate the queue manager, restart queue, and administration queue that are defined for a Q Capture schema.

APPLY SCHEMA

Specify to validate the queue manager that is defined for a Q Apply schema.

PUBQMAP

Specify to validate the send queue that is specified for a publishing queue map.

REPLQMAP

Specify to validate the send queue, receive queue, and Q Apply administration queue that are specified for a replication queue map.

QSUB

Specify to validate the model queue that is defined to create spill queues for a Q subscription.

Usage notes

Messages that describe the results of the tests are sent to the standard output (stdout).

Example 1

To validate the send queue, receive queue, and Q Apply administration queue that are specified for a replication queue map SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ ENVIRONMENT FOR REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```

Example 2

To validate the model queue that is specified for the Q Subscription EMPLOYEE0001 that uses the replication queue map SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ ENVIRONMENT FOR QSUB EMPLOYEE0001  
USING REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```

VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command

Use the VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP command to send test messages that validate the message flow between the WebSphere MQ queues that are specified for a replication queue map.

Syntax

```
▶▶—VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP—queue_map_name————▶▶
```

Parameters

queue_map_name

Specifies the name of an existing replication queue map.

Usage notes

The command puts a test message on the send queue and attempts to get the message from the receive queue. It also puts a test message on the Q Apply administration queue and attempts to get the message from the Q Capture administration queue. Messages that describe the results of the tests are sent to the standard output (stdout).

Example

To test the message flow between queues that are part of a replication queue map named SAMPLE_ASN_TO_TARGET_ASN:

```
VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP SAMPLE_ASN_TO_TARGET_ASN
```

Chapter 6. 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 208 demonstrates how you can combine event publishing commands to create an ASNCLP setup script.

Table 12 lists the ASNCLP commands for event publishing and links to topics that describe each command.

Table 12. ASNCLP commands for event publishing

If you want to ...	Use this command
Add a column to a publication	ALTER ADD COLUMN command
Change a publishing queue map	“ALTER PUBQMAP command” on page 214
Change a publication	“ALTER PUB command” on page 212
Create the control tables for the Q Capture program	CREATE CONTROL TABLES FOR command
Create a publishing queue map	“CREATE PUBQMAP command” on page 226
Create a publication	“CREATE PUB command” on page 222
Drop the control tables for the Q Capture program	DROP CONTROL TABLES ON command
Delete a publishing queue map	“DROP PUBQMAP command” on page 229
Delete a publication	“DROP PUB command” on page 228
List publications	“LIST PUBS command” on page 230
List publishing queue maps	“LIST PUBQMAPS command” on page 231
List Q Capture schemas	LIST CAPTURE SCHEMA command
Promote a publication	“PROMOTE PUB command” on page 232
Promote a publishing queue map	“PROMOTE PUBQMAP command” on page 234
Start a publication	“START PUB command” on page 244
Set the Q Capture schema for all task commands	SET CAPTURE SCHEMA command
Define the log file for the ASNCLP program	SET LOG command
Define output files that contain SQL statements to set up event publishing	SET OUTPUT command
Set the WebSphere MQ queue manager	SET QMANAGER command
Specify whether to automatically run each task command from an input file before the ASNCLP program processes the next task command	SET RUN SCRIPT command
Specify the Q Capture server to use in the ASNCLP session	SET SERVER command
Enable and disable the trace for the ASNCLP commands	SET TRACE command
Display the environment set during the session	SHOW SET ENV command
Stop a publication	“STOP PUB command” on page 244

Table 12. ASNCLP commands for event publishing (continued)

If you want to ...	Use this command
Verify that the required WebSphere MQ objects exist and have the correct properties for schemas, queue maps, and publications.	VALIDATE WSMQ ENVIRONMENT FOR command

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 a 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 13 on page 210 below the sample describes the SQL scripts that are generated.

For help creating the WebSphere MQ objects that are used in these scripts, see WebSphere MQ setup script generator 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

```
# 1 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 "passwd";
SET QMANAGER "QM1" FOR CAPTURE SCHEMA;
SET CAPTURE SCHEMA SOURCE EP1;
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_ASNI_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_ASNI_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 (publication)

This script generates SQL statements to create a 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 a 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 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 EP1;
SET OUTPUT CAPTURE SCRIPT "pub.sql";
SET RUN SCRIPT LATER;

# 2 Creating the publication
# This command generates SQL to create a 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 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 CAPTURE SCHEMA SOURCE EP1;

```

```

# 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 13 describes the files.

Table 13. SQL script files that are created by the sample ASNCLP scripts

Output file	Contains SQL to ...
epcontrol.sql	Create Q Capture control tables

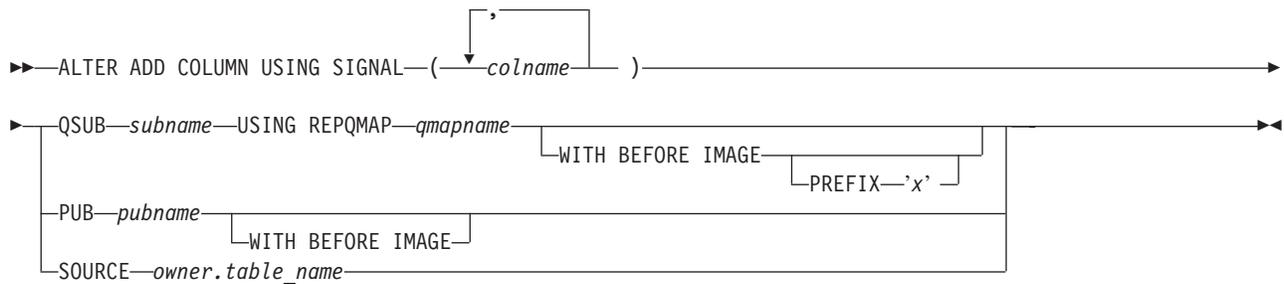
Table 13. SQL script files that are created by the sample ASNCLP scripts (continued)

Output file	Contains SQL to ...
pqmap.sql	Create a publishing queue map
pub.sql	Create a publication

ALTER ADD COLUMN command

Use the ALTER ADD COLUMN command to add a column to a Q subscription or publication.

Syntax



Parameters

colname

Specifies one or more columns (separated by a comma) to add to the definition of the active Q subscription or publication.

QSUB *subname*

Specifies the name of the Q subscription.

WITH BEFORE IMAGE

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, DB2 uses the default, X. If this prefix generates invalid names then other letters will be used beginning with Y until valid names are generated.

USING REPMAP *qmapname*

Specifies the name of the replication queue map used by the Q subscription.

PUB *pubname*

Specifies the name of the publication.

SOURCE *owner.table_name*

Specifies that the columns are added to all of the subscriptions and publications that subscribe to the source table.

Usage notes

- The column needs to exist in the source table already and should not be part of any existing Q subscription or publication.
- The Q subscription or publication must be active.

- The column must be nullable or have a default value on the source table.
- The column name on the target table will be implicitly named the same as the column name on the source table.
- For LONG VARCHAR or GRAPHIC types, the DATA CHANGES INCLUDE VARCHAR COLUMNS option must be enabled. VARCHAR COLUMNS are variable length character columns. The DATA CHANGES INCLUDE VARCHAR COLUMNS is an option set on the source table by altering the table's attributes using SQL.
- There is a limit of 20 columns that can be inserted into the statement.
- This command works for unidirectional replication and event publishing.

Example 1

To alter a Q subscription by adding columns PHONE and ADDRESS to the EMPLOYEE0001 Q subscription:

```
ALTER ADD COLUMN USING SIGNAL (PHONE, ADDRESS) QSUB EMPLOYEE0001
USING REPMAP SAMPLE_ASN_TO_TARGETDB_ASN
```

Example 2

To add the PHONE, ADDRESS, and EMAIL columns to all subscriptions and publications that subscribe to the EMPLOYEE table.

```
ALTER ADD COLUMN USING SIGNAL (PHONE, ADDRESS, EMAIL) SOURCE DB2ADMIN.EMPLOYEE
```

ALTER PUB command

Use the ALTER PUB command to change the properties of a publication.

Syntax

```

▶▶ ALTER PUB pubname FOR source_name
    [ source_owner ] [ DESC "description" ]
▶ [ PUBQMAP qmapname ] [ OPTIONS opt-clause ]

```

opt-clause:

```

| [ SEARCH CONDITION "search_cond" ] [ ALL CHANGED ROWS [ N ] ] [ BEFORE VALUES [ N ] ]
| [ CHANGED COLS ONLY [ Y ] ] [ SUPPRESS DELETES [ N ] ] [ TOPIC "topic" ]
|

```

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
```

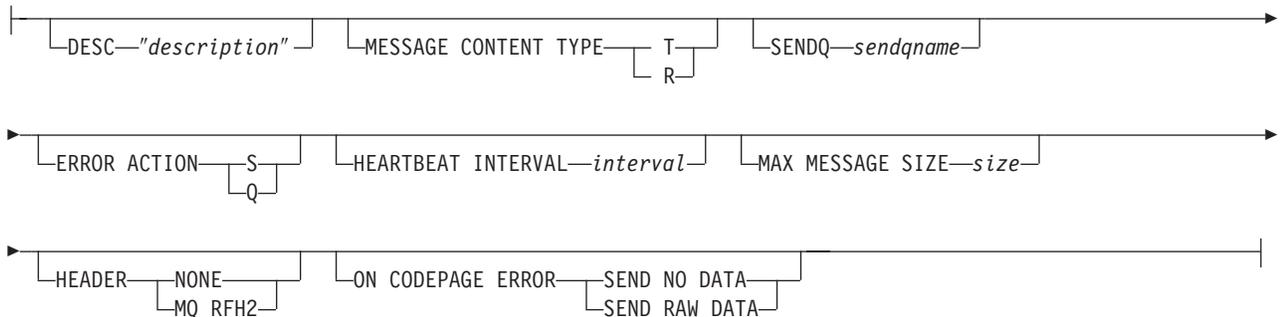
ALTER PUBQMAP command

Use the ALTER PUBQMAP command to change attributes for an existing publishing queue map.

Syntax

```
▶▶ ALTER PUBQMAP qmapname USING options |
```

options:



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

The action that the Q Capture program takes 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.

S The Q Capture program stops.

Q The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues.

HEARTBEAT INTERVAL *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 *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
```

ASNCLP SESSION SET TO command

Use the ASNCLP SESSION SET TO command to establish an ASNCLP session for Q replication to either relational or Classic data sources.

Syntax

▶▶—ASNCLP SESSION SET TO—Q REPLICATION—▶▶

Parameters

Q REPLICATION

Specify to set the ASNCLP session to Q replication. This ASNCLP session only accepts Q replication syntax. Use this parameter when you are connecting to either relational or Classic sources.

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.
- You can only issue commands that apply to the type of replication that you specify.

Example 1

To set the ASNCLP session to Q replication:

```
ASNCLP SESSION SET TO Q REPLICATION
```

CREATE CONTROL TABLES FOR command

Use the CREATE CONTROL TABLES FOR command to set up Q Capture and Q Apply control tables. For event publishing, Q Apply control tables are not needed.

For bidirectional and peer-to-peer replication, run the SET MULTIDIR SCHEMA command before you use this command. The Q Capture and Q Apply programs must use the same schema on each server.

In Classic replication, the control tables for the Classic capture components are creating by using the Classic Data Architect.

Syntax

```
▶▶ CREATE CONTROL TABLES FOR { CAPTURE SERVER | APPLY SERVER } USING capparms-clause
```

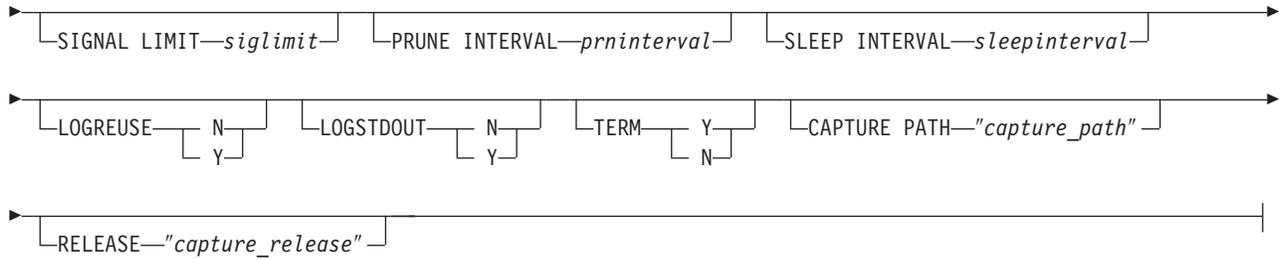
```
▶ { ZOS INDEX } zos-idx-clause
```

capparms-clause:

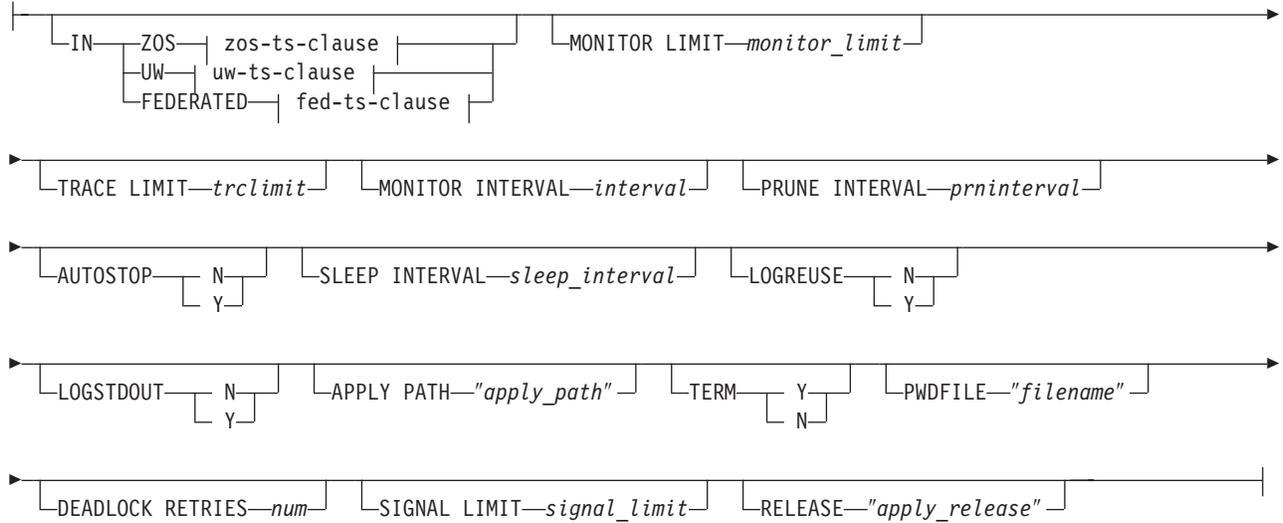
```
▶ { IN { ZOS } zos-ts-clause | { UW } uw-ts-clause } RESTARTQ "rstqname" ADMINQ "admname"
```

```
▶ { STARTMODE { WARM SI } | MEMORY LIMIT limit | AUTOSTOP { N } | { COLD } | { WARM NS } }
```

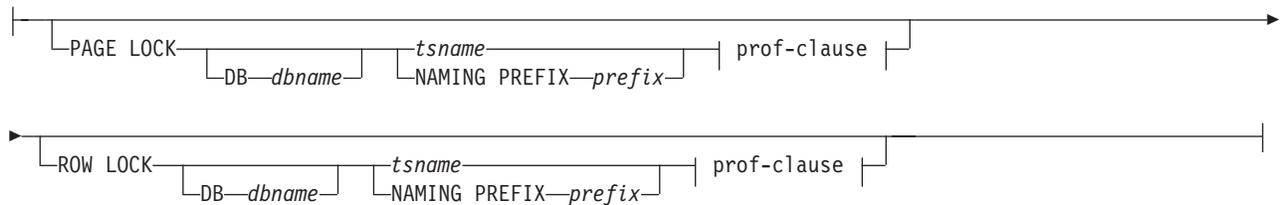
```
▶ { MONITOR INTERVAL interval | MONITOR LIMIT monlimit | TRACE LIMIT trclimit }
```



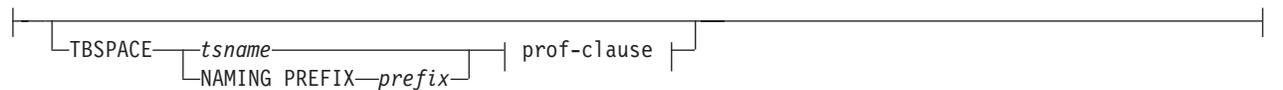
applyparms-clause:



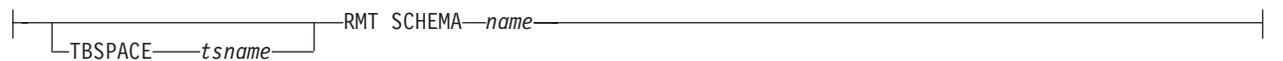
zos-ts-clause:



uw-ts-clause:



fed-ts-clause:



prof-clause:



zos-idx-clause:



Parameters

CAPTURE SERVER

Specify to create Q Capture control tables.

APPLY SERVER

Specify to create Q Apply control tables.

capparms-clause:

ZOS

Specifies a z/OS system on which to create Q Capture control tables.

UW

Specifies a Linux, UNIX, or Windows system on which to create Q Capture control tables.

RESTARTQ "rstqname"

Specifies the restart queue that the Q Capture program uses.

ADMINQ "admqname"

Specifies the administration queue that the Q Capture program uses.

STARTMODE

Specifies what kind of start the Q Capture program will perform.

WARMSI

Specify for the Q Capture program to perform a warm start. If the Q Capture program is starting for the first time, it will perform a cold start.

COLD

Specify for the Q Capture program to perform a cold start.

WARMNS

Specify for the Q Capture program to attempt a warm start if information is available. If the information is not available, the Q Capture program will stop.

MEMORY LIMIT *limit*

Specifies the maximum amount (in MB) of memory that the Q Capture program can use to build transactions.

AUTOSTOP

N The Q Capture or Q Apply program does not stop after it reaches the end of the active log and finds no transactions.

Y The Q Capture or Q Apply program stops after it reaches the end of the active log and finds no transactions.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Capture program inserts rows into the IBMQREP_CAPMON table.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPMON and IBMQREP_CAPQMON tables before it becomes eligible for pruning. All rows in these tables that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_CAPTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

SIGNAL LIMIT *siglimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_SIGNAL table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_CAPMON, IBMQREP_CAPQMON, IBMQREP_CAPTRACE, and IBMQREP_SIGNAL tables are pruned.

SLEEP INTERVAL *sleepinterval*

Specifies the number of milliseconds that the Q Capture program sleeps when it finishes processing the active log and determines that the buffer is empty.

LOGREUSE

- N The Q Capture program appends messages to the log file, even after the Q Capture program restarts.
- Y The Q Capture program reuses the log file by first truncating the current log file and then starting a new log when the Q Capture program restarts.

LOGSTDOUT

- N The Q Capture program only sends messages to the log file.
- Y The Q Capture program sends messages to both the log file and the standard output (stdout).

TERM

- Y The Q Capture program terminates if DB2 is quiesced or stops. This value is the default.
- N The Q Capture program continues running if DB2 is quiesced or stops.

CAPTURE_PATH *"capture_path"*

Specifies the location of the work files that the Q Capture program uses. On z/OS systems, the location can be an MVS data set high-level qualifier with //. The default is NULL.

Linux UNIX Windows RELEASE *"capture_release"*

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks ("). Specifying the release level enables newer replication and publishing function on an older DB2.

appparms-clause:

ZOS

Specifies a z/OS system in which Q Apply control tables are created.

UW

Specifies a Linux, UNIX, or Windows system in which Q Apply control tables are created.

FEDERATED

Specifies a federated target, on which Q Apply control tables are created in an Oracle, Sybase, Informix, Microsoft SQL Server, or Teradata database, and nicknames are created for these control tables in the Q Apply server. Some control tables are created in the Q Apply server.

MONITOR LIMIT *monlimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_APPLYMON table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

TRACE LIMIT *trclimit*

Specifies how long (in minutes) a row can remain in the IBMQREP_APPLYTRACE table before it becomes eligible for pruning. All rows that are older than the specified value are pruned at the next pruning cycle.

MONITOR INTERVAL *interval*

Specifies how frequently (in milliseconds) the Q Apply program inserts rows into the IBMQREP_APPLYMON table.

PRUNE INTERVAL *prninterval*

Specifies how frequently (in seconds) the IBMQREP_APPLYMON and IBMQREP_APPLYTRACE tables are pruned.

AUTOSTOP

- N The Q Apply program does not stop after all queues are emptied once.
- Y The Q Apply program stops after all queues are emptied once.

LOGREUSE

- N The Q Apply program appends messages to the log file, even after the Q Apply program is restarted.
- Y The Q Apply program reuses the log file by first truncating the current log file and then starting a new log when the Q Apply program is restarted.

LOGSTDOUT

- N The Q Apply program sends messages only to the log file.
- Y The Q Apply program sends messages to the log file and the standard output (stdout).

APPLY PATH "*apply_path*"

Specifies the location of the work files the Q Apply program uses. The default path is the directory where the asnqapp command was run.

TERM

- Y The Q Apply program stops if DB2 is quiesced or stops.
- N The Q Apply program continues running if DB2 is quiesced or stops.

PWDFILE "*filename*"

Specifies the name of the password file.

DEADLOCK RETRIES *num*

Specifies the number of retries for SQL deadlock errors.

Linux UNIX Windows **RELEASE** "*apply_release*"

Specifies the release level of the control tables that you want to create. Allowed values are 9.7, 9.5, and 9.1. This parameter is for Linux, UNIX, and Windows only. Enclose value in double quotation marks ("). Specifying the release level enables newer replication and publishing function on an older DB2.

zos-ts-clause:

PAGE LOCK

Specify for replication control tables that require page-level locking.

ROW LOCK

Specify for replication control tables that require row-level locking.

DB *dbname*

Specifies the name of the database that contains the table space where the control tables will be created.

tsname

Specifies the name of the table space for the z/OS control tables.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

uw-ts-clause:

TBSPACE

tsname

Specifies the name of the table space that is used for the control tables on Linux, UNIX, or Windows.

NAMING PREFIX *prefix*

Specifies a prefix to add to the name of the table space.

fed-ts-clause:

TBSPACE *tsname*

Specifies the name of an existing Oracle table space, Sybase segment, Informix dbspace, or Microsoft SQL Server file group that is used for the control tables. This parameter is not applicable for Teradata targets.

RMT SCHEMA

The remote schema that the Q Apply program uses to create control tables on the non-DB2 database. The default is the remote authorization ID.

CREATE

Specify to create a table space. When this parameter is used without the USING PROFILE keyword, the table space is assumed to exist and the control tables are created in this table space.

USING PROFILE *pname*

Specifies the name of a profile to use to customize the table space attributes.

Example 1

To create Q Apply control tables and to specify a monitor limit of 3 minutes and a trace limit of 9 minutes:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING MONITOR LIMIT 3 TRACE LIMIT 9
```

Example 2

To create Q Capture control tables:

```
CREATE CONTROL TABLES FOR CAPTURE SERVER USING  
RESTARTQ "ASN1.QM1.RESTARTQ" ADMINQ "ASN1.QM1.ADMINQ"
```

Example 3

To create Q Apply control tables for replication to an Oracle target with a remote authorization ID of ORACLE_ID:

```
CREATE CONTROL TABLES FOR APPLY SERVER IN FEDERATED RMT SCHEMA ORACLE_ID
```

Example 4

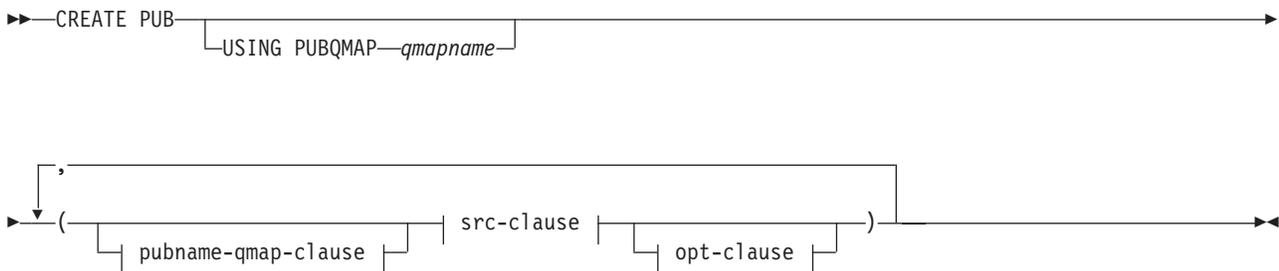
To create Version 9.7 Q Apply control tables on a DB2 Version 9.1 database:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING RELEASE "9.7"
```

CREATE PUB command

Use the CREATE PUB command to create a publication.

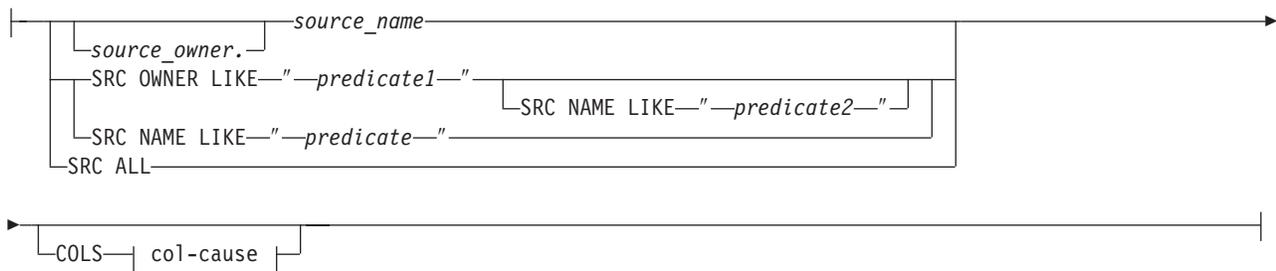
Syntax



pubname-qmap-clause:

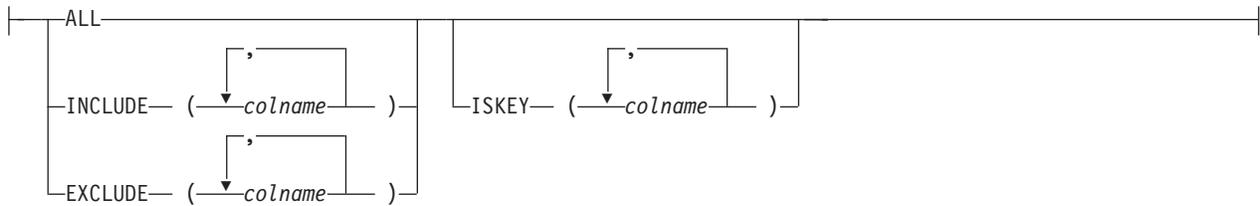


src-clause:

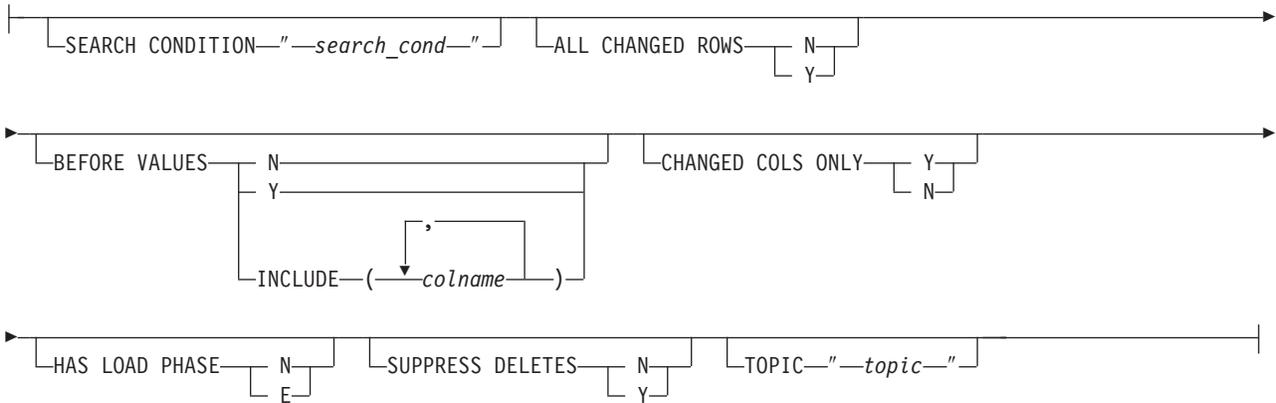


col-clause:





opt-clause:



Parameters

USING PUBQMAP *qmapname*

Specifies the publishing queue map that is used by all subsequent publications that are created by this command.

pubname-qmap-clause:

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.

src-clause:

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 (by 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 that uses 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;
```

```

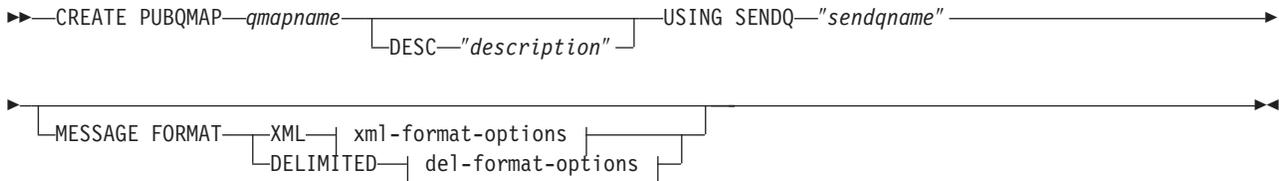
SET CAPTURE SCHEMA SOURCE SAMPLE;
CREATE PUB USING PUBQMAP 'PUBQ1' (PUBNAME PUB1 DATA.EMPLOYEE
OPTIONS BEFORE VALUES INCLUDE(c10, c11, c12));

```

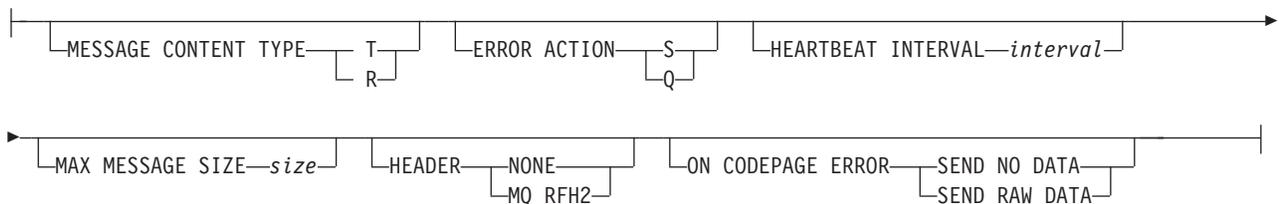
CREATE PUBQMAP command

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.

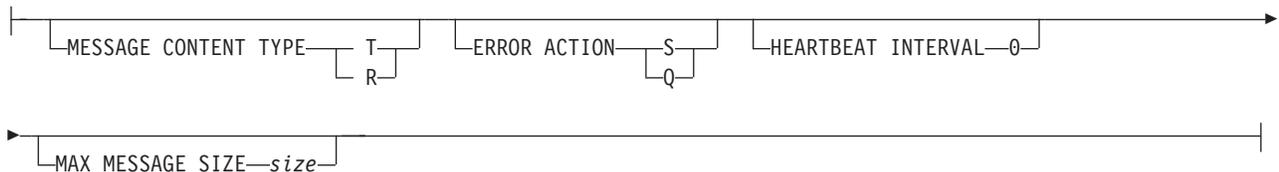
Syntax



xml-format-options:



del-format-options:



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

The action that the Q Capture program takes 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.

- S** The Q Capture program stops.
- Q** The Q Capture program stops putting messages on any send queues that are in error and continues putting messages on other send queues.

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
```

DROP CONTROL TABLES ON command

Use the `DROP CONTROL TABLES ON` command to drop the Q Capture and Q Apply control tables. In Classic replication, you can use this command to drop only the Q Apply control tables.

Syntax

```
▶▶ DROP CONTROL TABLES ON {CAPTURE SERVER | APPLY SERVER} ▶▶
```

Parameters

CAPTURE SERVER

Specify to drop the Q Capture control tables.

APPLY SERVER

Specify to drop the Q Apply control tables.

Usage notes

This command is used in conjunction with the `SET SERVER` command to indicate the location of the control tables.

Example

To drop the Q Capture control tables:

```
SET SERVER TARGET TO QAPPDB;
DROP CONTROL TABLES ON APPLY SERVER
```

DROP PUB command

Use the `DROP PUB` command to delete a publication.

Syntax

```
▶▶ DROP PUB {ALL | (—PUBNAME—pubname—) | FOR PUBNAME LIKE—"predicate"—} ▶▶
```

Parameters

ALL

Specify to delete all of the publications for the schema and server set through the SET commands.

PUBNAME *pubname*

Specifies the name of a 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)
```

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
```

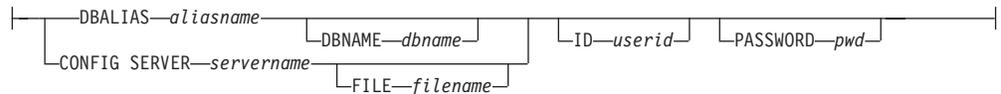
LIST CAPTURE SCHEMA command

You can use the LIST CAPTURE SCHEMA command to list the Q Capture schemas for a specified server.

Syntax

```
▶▶—LIST CAPTURE SCHEMA—  
└──SERVER──| dbparms |──▶▶
```

dbparms-clause:



Parameters

dbparms-clause:

SERVER

Specifies the server that contains the schemas to be listed.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

Example

To list the Q Capture schema on server SAMPLE:

```
LIST CAPTURE SCHEMA SERVER DBALIAS SAMPLE ID id1 PASSWORD "passwd!";
```

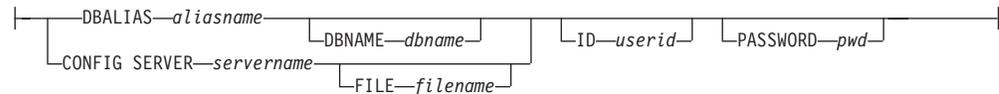
LIST PUBS command

You can use the LIST PUBS command to list publications for a specified Q Capture server or schema.

Syntax



dbparms-clause:



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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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.

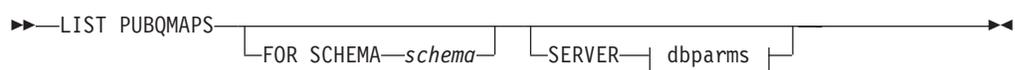
FILE *filename*

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 `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different environments.

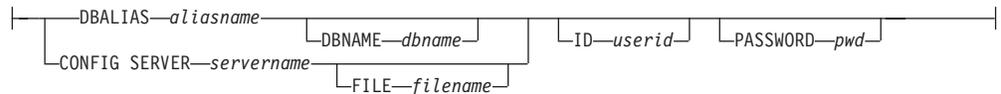
LIST PUBQMAPS command

You can use the LIST PUBQMAPS command to list publication queue maps for a specified Q Capture server or schema.

Syntax



dbparms-clause:



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 of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, or Windows database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

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*

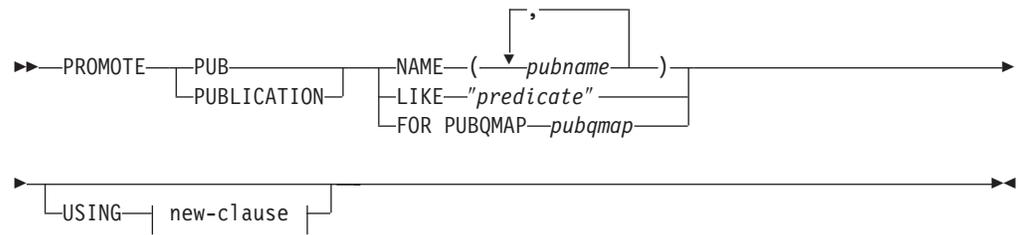
Specifies the complete path and file name to the replication configuration file. If you do not use the **FILE** parameter, the ASNCLP attempts to use the `asnservers.ini` file in the current directory, if that file exists. Use the **FILE** parameter with different files that are customized for different 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 publishing queue map that it uses. 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



new-clause::



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 that use the publishing queue map are promoted.

new-clause:

USING PUBQMAP *newqmap*

Specifies the name of a new publishing queue map that you want to use for the promoted publications.

Example - matching a predicate

To promote all publications that start with the name EMP:

```
PROMOTE PUBLICATION LIKE "EMP%";
```

Example - using a publishing queue map

To promote all publications that use the qmap1 publishing queue map :

```
PROMOTE PUBLICATION FOR PUBQMAP qmap1;
```

Example - changing to a new publishing queue map

To promote all publications that use the publishing queue map qmap1 so that they use the queue map qmap2 instead:

```
PROMOTE PUBLICATION FOR PUBQMAP qmap1 USING PUBQMAP "qmap2";
```

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 USING new-clause | LIKE "predicate" ➤➤
```

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:

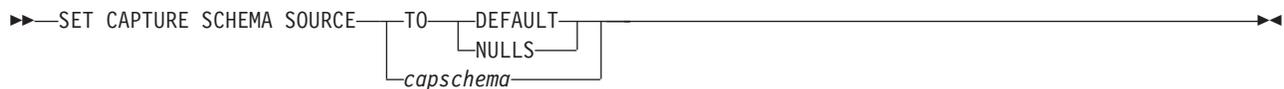
```
PROMOTE PUBQMAP NAME PUBQMAP2 USING PUBQMAP pubqmapnew SENDQ "sendqnew2";
```

SET CAPTURE SCHEMA command

Use the SET CAPTURE SCHEMA command to set a default schema of the source control tables for all task commands. For Classic sources, you can use only the default Q Capture schema, ASN.

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

Syntax



Parameters

SOURCE

Specifies the Q Capture schema. If you are using a DB2 source, the schema can be any valid DB2 schema name. If you are using a Classic source, you must use the DEFAULT schema.

DEFAULT

Specify to set the Q 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 Q Capture schema name.

Example 1

To reset the default Q Capture schema to ASN:

```
SET CAPTURE SCHEMA SOURCE TO DEFAULT
```

Example 2

To set the default Q Capture schema to ASN1:

```
SET CAPTURE SCHEMA SOURCE ASN1
```

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—"logfile"—▶▶
```

Parameters

"logfile"

Specifies the output log file name. The default log file name is `qrep1msg.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 `qmaplog.err` for creating replication queue maps:

```
SET LOG "qmaplog.err"
```

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, or the ASNCLP commands needed to promote a replication environment. You cannot use this command with non-relational sources.

Syntax

```
▶▶—SET OUTPUT—  
┌CAPTURE SCRIPT—"capfname"┐ ┌TARGET SCRIPT—"trgfname"┐  
▶▶  
└PROMOTE SCRIPT—"profname"└  
▶▶
```

Parameters

CAPTURE SCRIPT "*capfname*"

Specifies the output file name for SQL scripts that run at the Q Capture server.

TARGET SCRIPT "*trgfname*"

Specifies the output file name for SQL scripts that run at the Q Apply, or target server.

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 `qrep1_asnc1p.in`.

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 target script output file "target.sql":

```
SET OUTPUT TARGET SCRIPT "target.sql"
```

SET QMANAGER command

Use the SET QMANAGER command to set the WebSphere MQ queue manager. You cannot use this command with non-relational sources.

Syntax

```
▶▶—SET QMANAGER—"mgrname"—FOR—

|                                        |
|----------------------------------------|
| CAPTURE SCHEMA                         |
| APPLY SCHEMA                           |
| MULTIDIR— <i>servername.schemaname</i> |

—▶▶
```

Parameters

"mgrname"

Specifies the name of the WebSphere MQ queue manager.

CAPTURE SCHEMA

Specify to set the queue manager for the Q Capture control tables.

APPLY SCHEMA

Specify to set the queue manager for the Q Apply control tables.

MULTIDIR

Specify to set the queue manager for the bidirectional or peer-to-peer replication server.

servername

Specifies the name of the server (database).

schemaname

Specifies the schema of the control tables.

Example 1

To set the queue manager QM1 for the Q Capture program:

```
SET QMANAGER "QM1" FOR CAPTURE SCHEMA
```

Example 2

To set the queue manager QM2 for the Q Apply program:

```
SET QMANAGER "QM2" FOR APPLY SCHEMA
```

Example 3

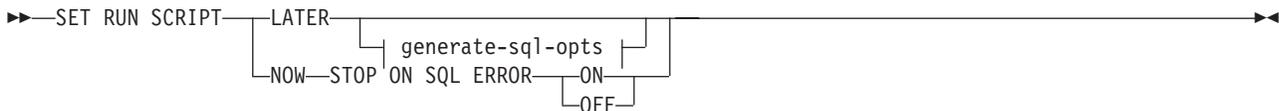
To set the queue manager QM3 for a server TESTDB.BLUE that is used in bidirectional or peer-to-peer replication:

```
SET QMANAGER "QM3" FOR MULTIDIR TESTDB.BLUE
```

SET RUN SCRIPT command

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. You cannot use the LATER parameter with non-relational sources.

Syntax



generate-sql-opts:



Parameters

LATER

Specify to run the SQL scripts at a later time. You cannot use this parameter with Classic sources. Use this option if you want to verify your script before you run it. You can also use this option if you want to create SQL script files on one operating system, but run them on another.

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.

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. You cannot use this parameter with Classic sources.

GENERATE SQL FOR EXISTING

Specifies whether to generate SQL when ASNCLP encounters errors due to duplicate, or existing, objects when processing CREATE commands. This option has no effect on DROP commands.

NO

The ASNCLP program will not generate SQL to create objects that already exist. This is the default.

YES

The ASNCLP program continues to generate SQL statements even if it encounters existing object errors. The following errors are ignored when you specify this option:

Create Control Tables

Another control table already exists under the same schema or table spaces are specified to be created but they already exist.

Create Publication

Another publication with the same name already exists.

Create Publishing Queue Map

Another publishing queue map with the same name already exists.

Create Replication Queue Map

Another replication queue map with the same name already exists.

Create Subscription

Another Q Subscription with the same name already exists, a target table already exists but the option in the CREATE QSUB command is to create the target table, the target table already exists but the option to create the table space was specified, or a unique index with the same name already exists.

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 Q subscriptions or publications 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 might 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 SCRIPT NOW be used.

Figure 2 on page 140 shows these dependencies for Q replication to a relational source. This figure does not apply to non-DB2 sources.

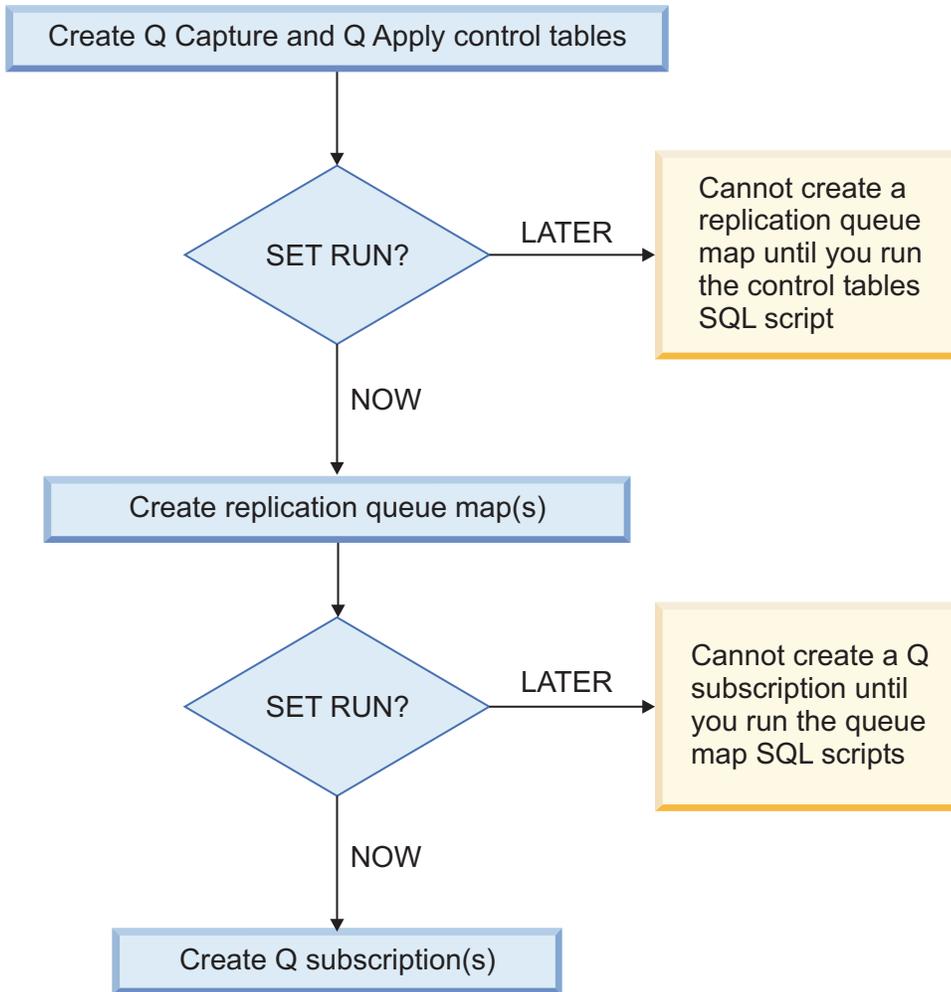


Figure 3. Dependencies between ASNCLP commands for Q replication from a DB2 source. This diagram shows the dependencies between ASNCLP CREATE commands that are used to set up Q replication. It assumes all objects use the default schema of ASN. The dependencies for Q Capture controls tables, publishing queue maps, and publications that are used in event publishing are the same.

Example - Run immediately and stop on errors

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
```

Example - Create SQL script and ignore errors when creating existing objects

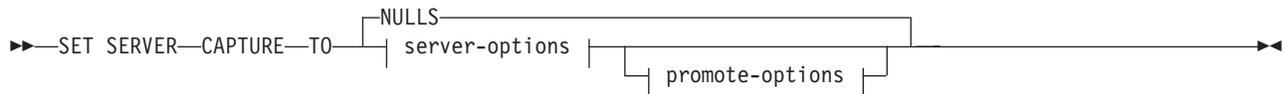
To generate the SQL scripts instead of running them immediately, and to continue generating SQL when creating objects that already exist:

```
SET RUN SCRIPT LATER GENERATE SQL FOR EXISTING YES;
```

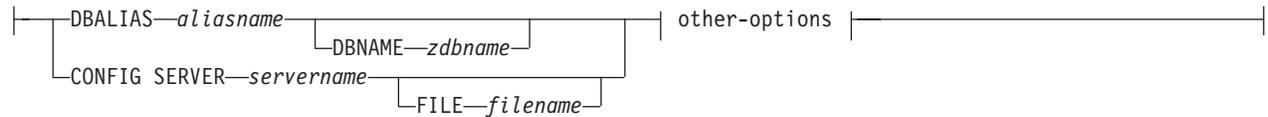
SET SERVER command (event publishing)

Use the SET SERVER command to specify the Q Capture server to use in the ASNCLP session. After you set a server name, all subsequent commands in the session apply to this server until you change the server with this command.

Syntax



server-options:



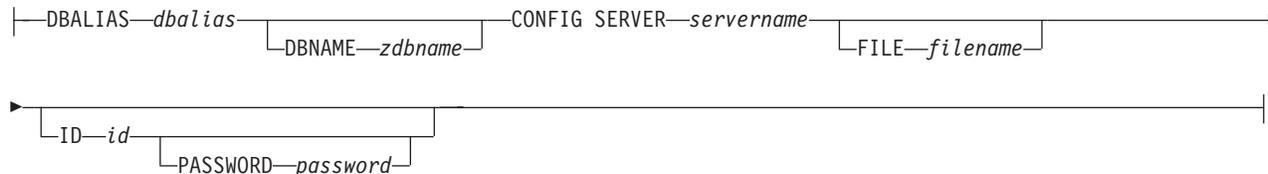
other-options:



promote-options:



promote-srvr-options:



Parameters

CAPTURE

Specify to set the database as a Q Capture or Classic server.

NULLS

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

server-options:

DBALIAS *aliasname*

Specifies the database alias name.

z/OS

DBNAME *zdbname*

Specifies the database name.

CONFIG SERVER *servername*

Classic sources: Specifies the Classic source that the ASNCLP program connects to. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the **FILE** parameter, the ASNCLP program attempts to use the `asnservers.ini` file in the current directory, if that file exists.

other-options:

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.

promote-options:

PROMOTE TO

Promote the specified server definitions.

SCHEMA *promoteschema*

Specifies the schema under which the server definitions will be promoted. If a schema is not specified, then the schema under which the current server definitions exist is used.

promote-srvr-options:

DBALIAS *dbalias*

Specifies the database that will receive the promoted server definitions. If this clause is not specified and a **PROMOTE** command is included in the input file, then the **PROMOTE** command promotes the definitions to the current server.

z/OS

DBNAME *zdbname*

Specifies the name of the database subsystem that will receive the promoted definitions.

CONFIG SERVER *servername*

Specifies the replication target that the ASNCLP program connects to when promoting definitions. The server name must match the bracketed `[NAME]` field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the **FILE** parameter, the ASNCLP program attempts to use the `asnservers.ini` file in the current directory, if that file exists.

ID *id*

Specifies the database ID where definitions will be promoted to. If not specified, the ASNCLP output script is generated without ID information.

PASSWORD *password*

Specifies the password to use to connect to the database. If not specified, the ASNCLP output script is generated without password information.

Example

To set the Q Capture server to the database SAMPLE:

```
SET SERVER CAPTURE TO DBALIAS SAMPLE;
```

Example - Classic sources

Given a configuration file called `classic.ini` that contains the following information:

```
[classic1]
Type=CLASSIC
Data source=CACSAMP
Host=9.30.155.156
Port=8019
```

Use the following command to specify server `classic1` as the data server:

```
SET SERVER CAPTURE TO CONFIG SERVER classic1 FILE classic.ini ID id1 PASSWORD pwd1;
```

Example - password prompting

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

```
SET SERVER CAPTURE 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. Your text is hidden when you type.

Example - promoting configurations

To set the existing server containing definitions to be promoted and set the new server that will receive these promoted configurations:

```
SET SERVER CAPTURE TO DBALIAS SAMPLE ID id1 PASSWORD "p1wd"
PROMOTE TO DBALIAS SAMPLE1 ID id1 PASSWORD SCHEMA ASN;
```

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

- All output is sent to the console. For readability, save the output to a file.

Example

To turn on the internal trace for the ASNCLP program:

```
SET TRACE ON
```

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 MYPUB
```

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
```

VALIDATE WSMQ ENVIRONMENT FOR command

Use the VALIDATE WSMQ ENVIRONMENT FOR command to verify that the required WebSphere MQ objects exist and have the correct properties for Q replication schemas, queue maps, and Q subscriptions.

Chapter 7. 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 248 demonstrates how you can combine Replication Alert Monitor commands to create an ASNCLP setup script.

Table 14 lists the ASNCLP commands for the Replication Alert Monitor and links to topics that describe each command.

Table 14. ASNCLP commands for the Replication Alert Monitor

If you want to ...	Use this command
Change alert conditions for the Apply program	“ALTER ALERT CONDITIONS FOR APPLY command” on page 251
Change alert conditions for the Capture program	“ALTER ALERT CONDITIONS FOR CAPTURE command” on page 253
Change alert conditions for the Q Apply program	“ALTER ALERT CONDITIONS FOR QAPPLY command” on page 256
Change alert conditions for the Q Capture program	“ALTER ALERT CONDITIONS FOR QCAPTURE command” on page 258
Change contact information for notifications	“ALTER CONTACT command” on page 260
Change a contact group	“ALTER GROUP command” on page 261
Change a monitor suspension	“ALTER MONITOR SUSPENSION command” on page 261
Change a monitor suspension template	“ALTER MONITOR SUSPENSION TEMPLATE command” on page 262
Create alert conditions for the Apply program	“CREATE ALERT CONDITIONS FOR APPLY command” on page 263
Create alert conditions for the Capture program	“CREATE ALERT CONDITIONS FOR CAPTURE command” on page 266
Create alert conditions for the Q Apply program	“CREATE ALERT CONDITIONS FOR QAPPLY command” on page 267
Create alert conditions for the Q Capture program	“CREATE ALERT CONDITIONS FOR QCAPTURE command” on page 269
Create contact information for notifications	“CREATE CONTACT command” on page 271
Create the control tables for the Monitor program	“CREATE CONTROL TABLES FOR command” on page 272
Create a contact group	“CREATE GROUP command” on page 274
Create a monitor suspension	“CREATE MONITOR SUSPENSION command” on page 274
Create a monitor suspension template	“CREATE MONITOR SUSPENSION TEMPLATE command” on page 276
Delegate an existing contact to a new contact	“DELEGATE CONTACT command” on page 277

Table 14. ASNCLP commands for the Replication Alert Monitor (continued)

If you want to ...	Use this command
Delete alert conditions for the Apply program	"DROP ALERT CONDITIONS FOR APPLY command" on page 277
Delete alert conditions for the Capture program	"DROP ALERT CONDITIONS FOR CAPTURE command" on page 278
Delete alert conditions for the Q Apply program	"DROP ALERT CONDITIONS FOR QAPPLY command" on page 278
Delete alert conditions for the Q Capture program	"DROP ALERT CONDITIONS FOR QCAPTURE command" on page 278
Delete an existing contact	"DROP CONTACT command" on page 279
Delete a contact group	"DROP GROUP command" on page 279
Delete a monitor suspension	"DROP MONITOR SUSPENSION command" on page 280
Delete a monitor suspension template	"DROP MONITOR SUSPENSION TEMPLATE command" on page 280
List monitor suspensions	"LIST MONITOR SUSPENSION command" on page 281
List monitor suspension templates	"LIST MONITOR SUSPENSION TEMPLATE command" on page 281
Specify the server (database) used in the ASNCLP session, authentication information, and other required parameters for connecting to the server	"SET SERVER command" on page 282
Substitute one existing contact with another existing contact	"SUBSTITUTE CONTACT command" on page 283

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 15 on page 251 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:

- 1** Setting the environment
- 2** Creating Monitor control tables
- 3** Ending the ASNCLP session

```
# 1 Setting the environment.
# 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;
```

```
# 2 Creating Monitor control tables.
```

```
CREATE CONTROL TABLES FOR MONITOR CONTROL SERVER
IN UW OTHERS TSMON1;
```

```
# 3 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

```
# 1 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;
```

```
# 2 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";
```

```
# 3 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. The
# ASNCLP SESSION SET command is needed because the alert conditions are for
# Q replication programs.
```

```
ASNCLP SESSION SET TO Q REPLICATION;
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 15 on page 251 describes the files and where they run.

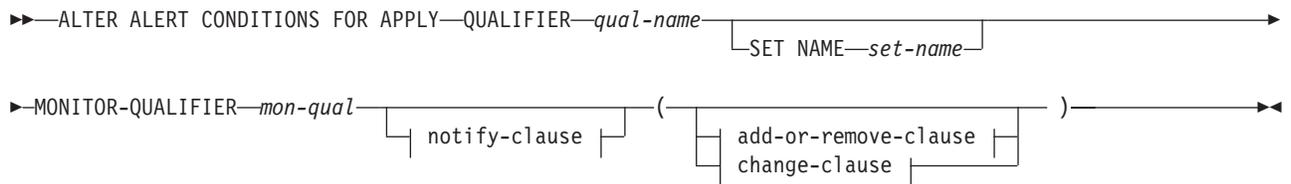
Table 15. SQL script files that are created by the sample ASNCLP scripts

Output file	Contains SQL to ...
moncontrol.sql	Create Monitor control tables
conalert.sql	Define a contact and alert conditions
suspend.sql	Create a suspension template and suspension

ALTER ALERT CONDITIONS FOR APPLY command

Use the ALTER ALERT CONDITIONS FOR APPLY command to alter alert conditions for the Apply program.

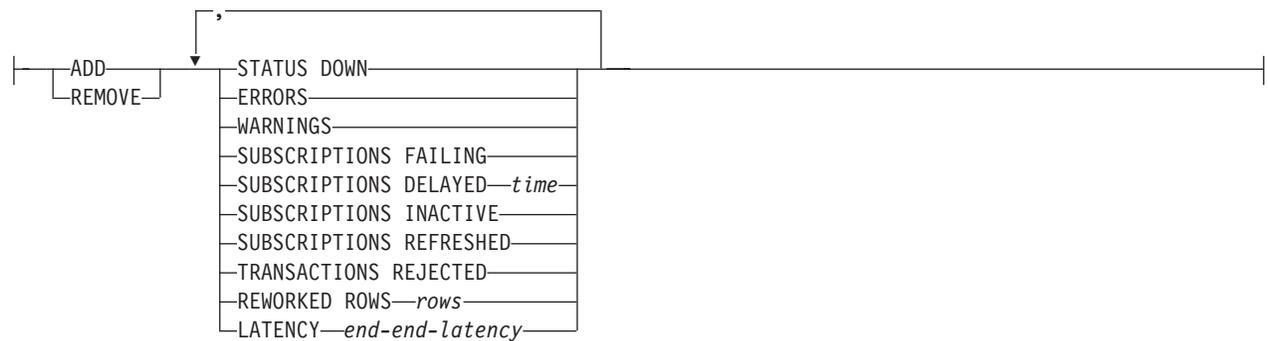
Syntax



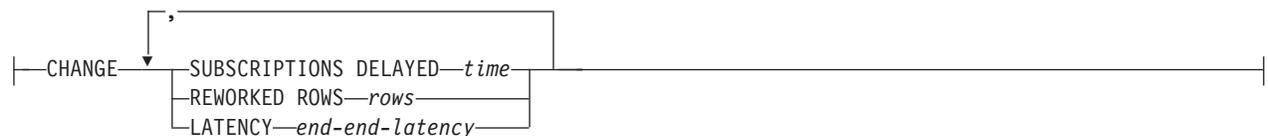
notify-clause:



add-or-remove-clause:



change-clause:



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 `asnacmd` status command to verify that the Apply program is running. The `asnacmd` 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: $(LAST_RUN + \text{user threshold in seconds} > \text{CURRENT TIMESTAMP})$.

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

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 alter an alert condition for the Apply program by removing the condition `WARNINGS` and no longer alerting the contact `REPLADMIN` when the condition occurs:

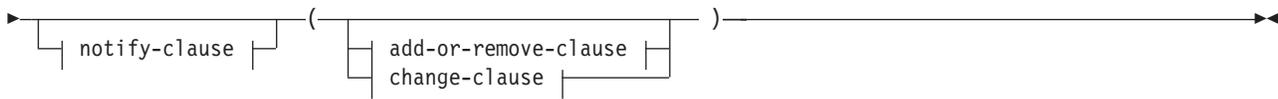
```
ALTER ALERT CONDITIONS FOR APPLY QUALIFIER MYAPPLY01 MONITOR QUALIFIER MONQUAL
NOTIFY REPLADMIN (REMOVE WARNINGS)
```

ALTER ALERT CONDITIONS FOR CAPTURE command

Use the `ALTER ALERT CONDITIONS FOR CAPTURE` command to alter alert conditions for the Capture program.

Syntax

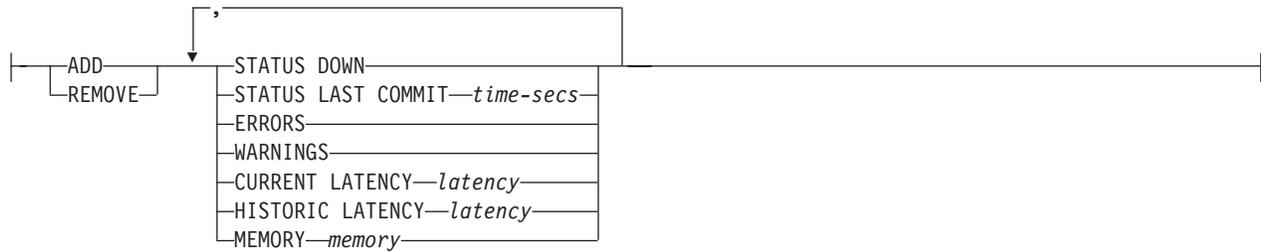
```
▶▶—ALTER ALERT CONDITIONS FOR CAPTURE—SCHEMA—cap-schema—MONITOR-QUALIFIER—mon-qual—▶▶
```



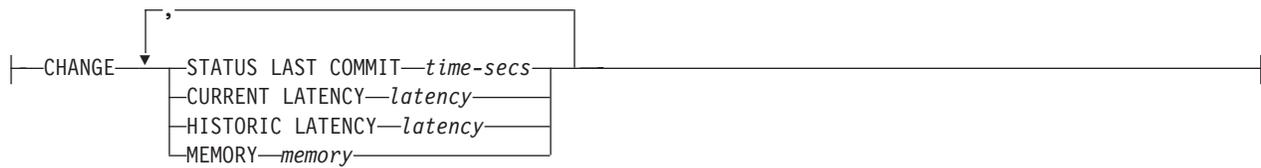
notify-clause:



add-or-remove-clause:



change-clause:



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

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 `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 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 by 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 by 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.

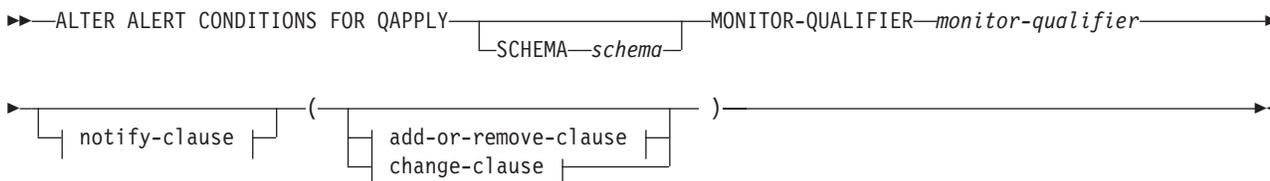
Example

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.

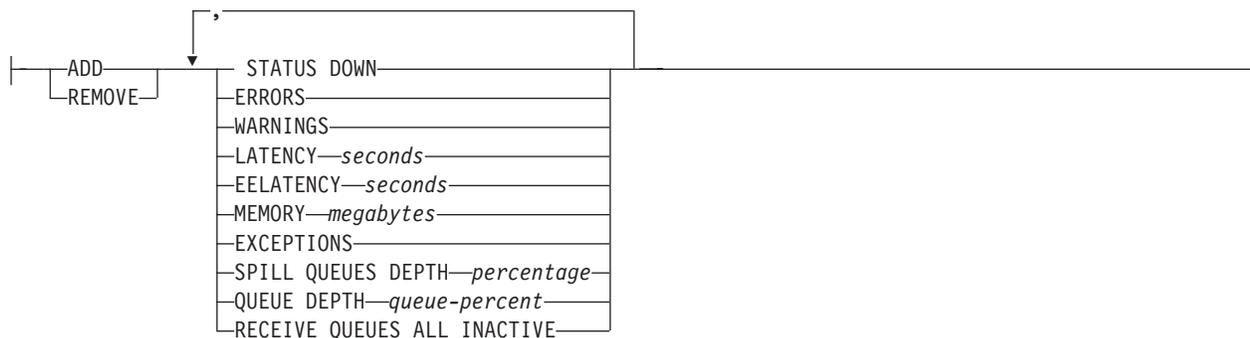
Syntax



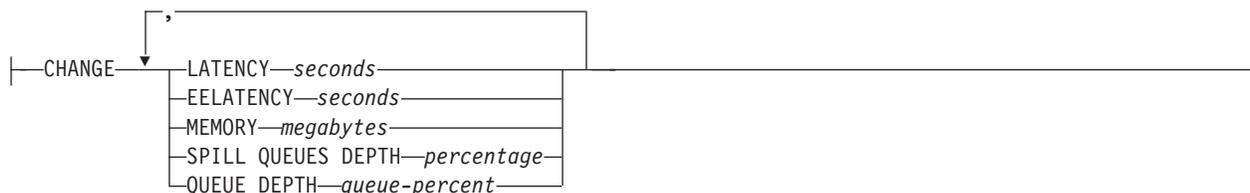
notify-clause:



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 `asnlcmd` 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

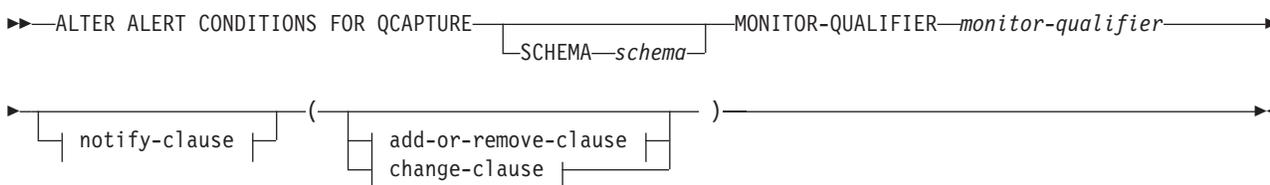
To alter an alert condition for the Q Apply program by removing the condition EXCEPTIONS and no longer alerting the contact REPLADMIN when the condition occurs:

```
ALTER ALERT CONDITIONS FOR QAPPLY MONITOR QUALIFIER MONQUAL
NOTIFY REPLADMIN (REMOVE EXCEPTIONS)
```

ALTER ALERT CONDITIONS FOR QCAPTURE command

Use the ALTER ALERT CONDITIONS FOR QCAPTURE command to alter the alert conditions for the Q Capture program.

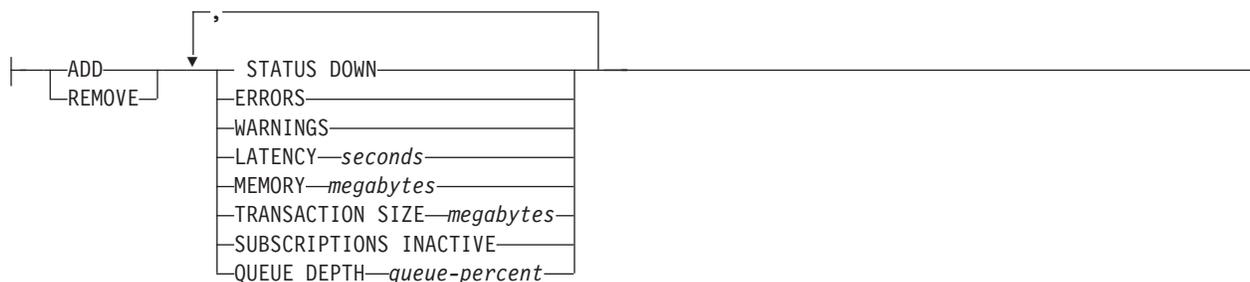
Syntax



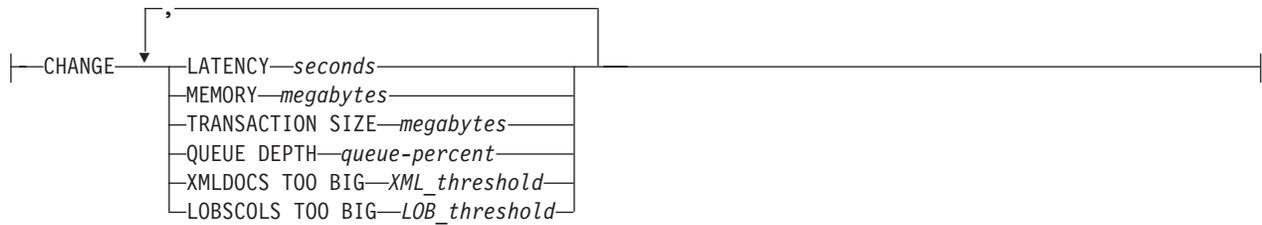
notify-clause:



add-or-remove-clause:



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 `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.

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 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

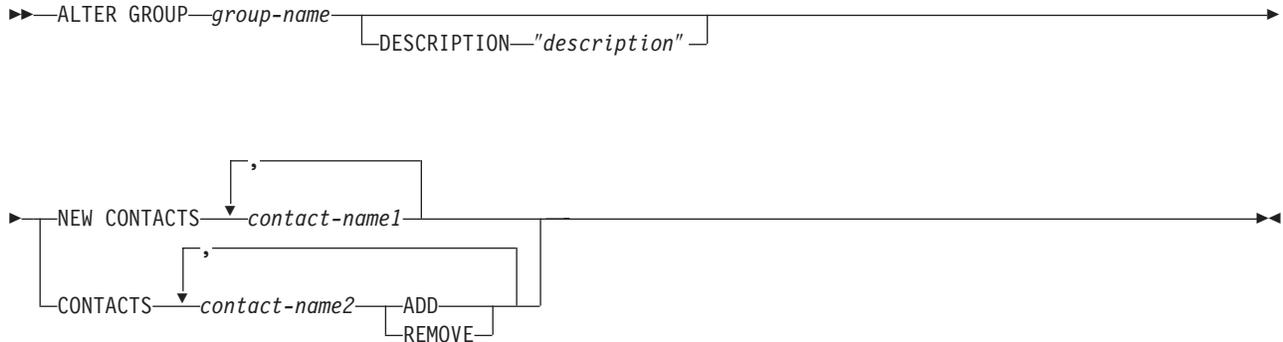
To alter a contact REPLADMIN by changing the e-mail address to repladmin@ibm.com:

```
ALTER CONTACT REPLADMIN EMAIL "repladmin@ibm.com"
```

ALTER GROUP command

Use the ALTER GROUP command to alter a group of replication monitor contacts.

Syntax



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:

```
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



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

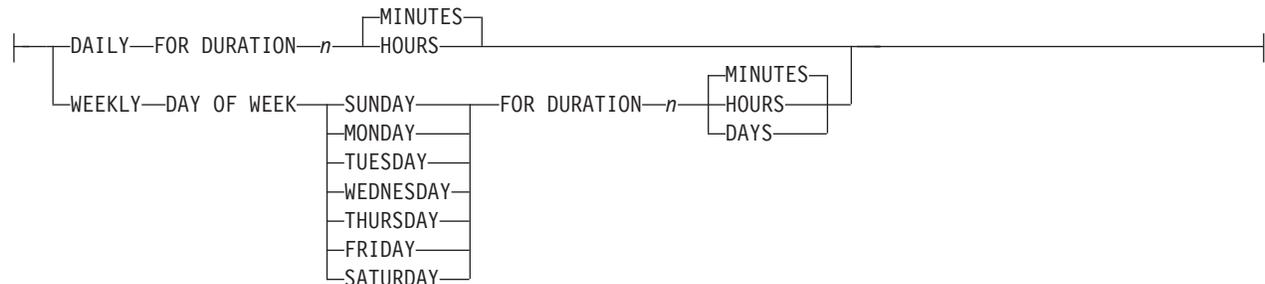
Use the `ALTER MONITOR SUSPENSION TEMPLATE` command to change the frequency and duration of periods that the monitor program is suspended.

Syntax

▶ ALTER MONITOR SUSPENSION TEMPLATE *template_name* [START TIME *HH:MM:SS*]

▶ REPEATS occurrence-clause

occurrence-clause:



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

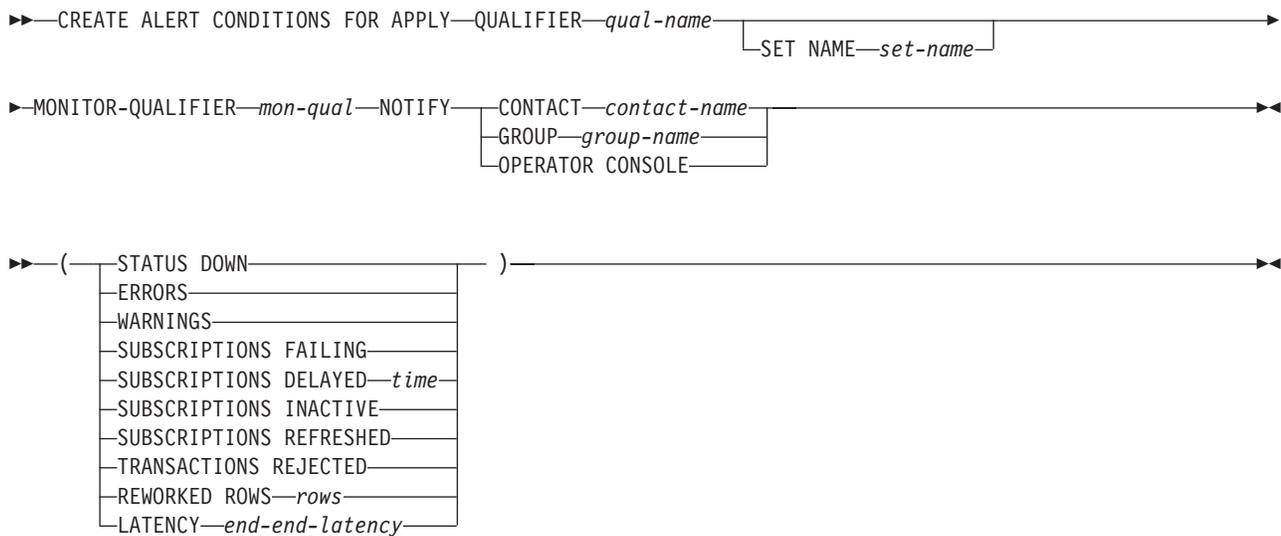
To lengthen a template that suspends the monitor program during the lunch hour every day to 90 minutes:

```
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



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.

STATUS DOWN

Specifies whether the Monitor program uses the `asnacmd` status command to verify that the Apply program is running. The `asnacmd` 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 *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

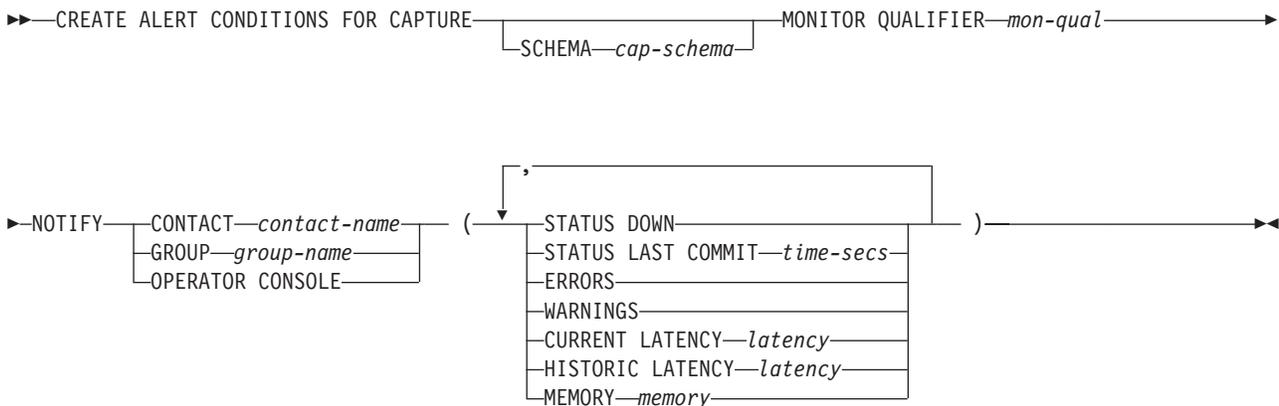
To create alert conditions for the Apply program that sends an alert to the contact REPLADMIN when a condition occurs:

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



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

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.

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 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 by 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 by 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

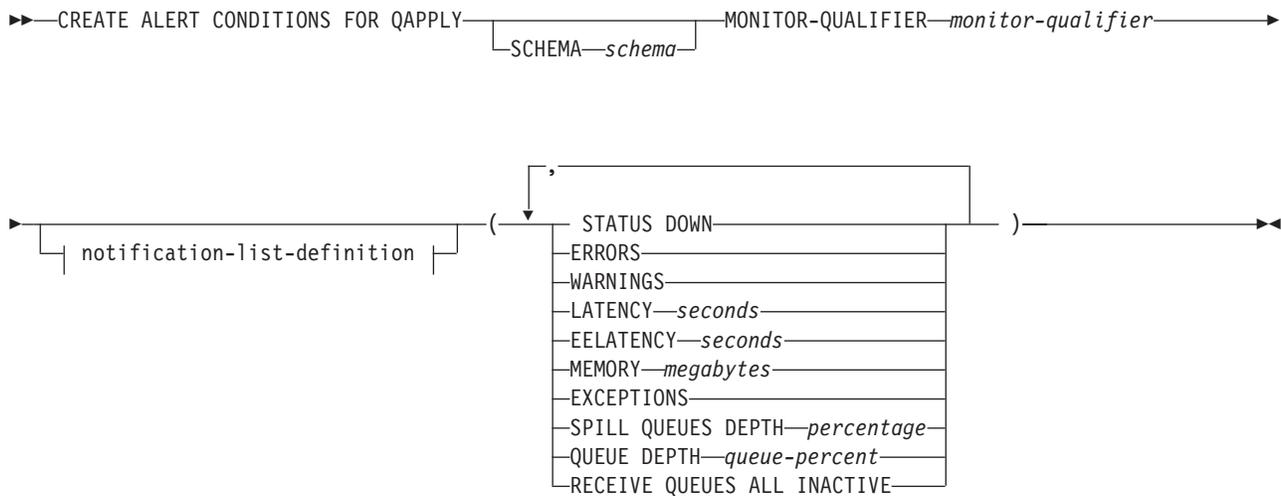
To create alert conditions for the Capture program that sends an alert to the contact REPLADMIN when a condition occurs:

```
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



notification-list-definition:



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 `asnqcmd` 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.

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 Apply program that send an alert to the contact REPLADMIN when a condition occurs:

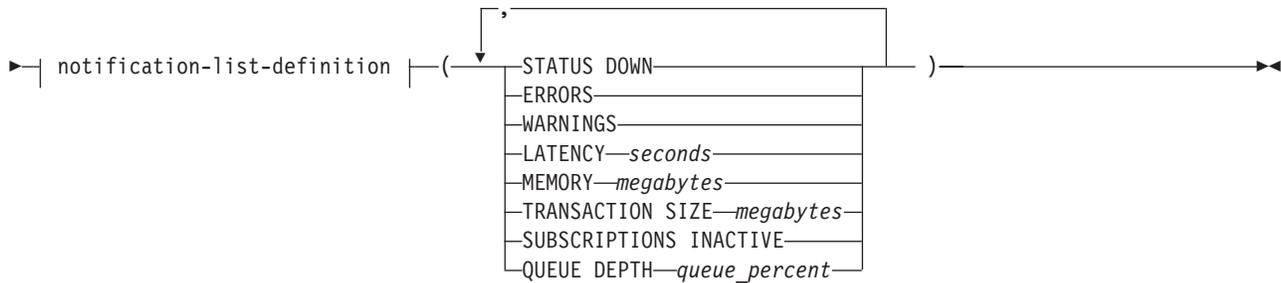
```
CREATE CONDITIONS FOR QAPPLY MONITOR QUALIFIER MONQUAL
  NOTIFY CONTACT REPLADMIN (STATUS DOWN, ERRORS, WARNINGS,
  LATENCY 360, EXCEPTIONS)
```

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:



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 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 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

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.

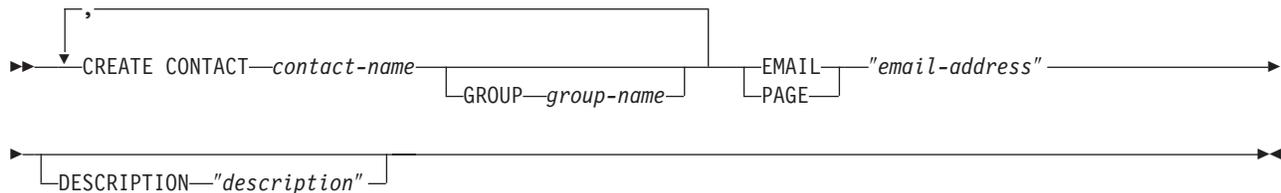
Example

To create alert conditions for the Q 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**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  
    UW uw-ts-clause  
    NONIBM fed-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
```

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

z/OS Specifies z/OS or OS/390.

UW

Linux UNIX Specifies UNIX or Windows.

NONIBM

Specifies non-DB2 data sources.

ALERTS

z/OS 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*

z/OS 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 and use 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 ─ [ MINUTES ]
└ ────┬─── [ HOURS ] ────┬──────────────────────────────────────────────────────────┘
      │
      └ WEEKLY ─ DAY OF WEEK ─ [ SUNDAY ] ─ FOR DURATION n ─ [ MINUTES ]
                            │   │   │   │   │   │   │   │   │   │   │   │   │   │
                            └─ MONDAY ─┘ ────┬─── [ HOURS ] ────┬──────────────────────────────────────────────────────────┘
                                                    │   │   │
                                                    └─ TUESDAY ─┘
                                                    └─ WEDNESDAY ─┘
                                                    └─ THURSDAY ─┘
                                                    └─ FRIDAY ─┘
                                                    └─ SATURDAY ─┘

```

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 12:00:00 REPEATS DAILY
FOR DURATION 1 HOUR

DELEGATE CONTACT command

Use the DELEGATE CONTACT command to delegate an existing contact to a new contact for a specific period of time.

Syntax

►► 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 (PERFORMANCE) for a given period of time:

```
DELEGATE CONTACT REPLADMIN TO PERFORMANCE 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

►► DROP ALERT CONDITIONS FOR APPLY QUALIFIER *apply-qual* MONITOR QUALIFIER *mon-qual* ◀◀

Parameters

APPLY QUALIFIER *qual-name*

Specifies the Apply qualifier.

MONITOR QUALIFIER *mon-qual*

Specifies the Monitor qualifier.

Example

To drop alert conditions for the Apply program:

```
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:

TEMPLATE_NAME	START_TIME	FREQUENCY	DURATION	UNITS
-----	-----	-----	-----	-----
daytemp1	12:00:00	DAILY	4	HOURS
wednesdaytemp2	00:00:00	WEDNESDAY	2	DAYS
minutestemp3	17:30:00	SUNDAY	30	MINUTES

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":

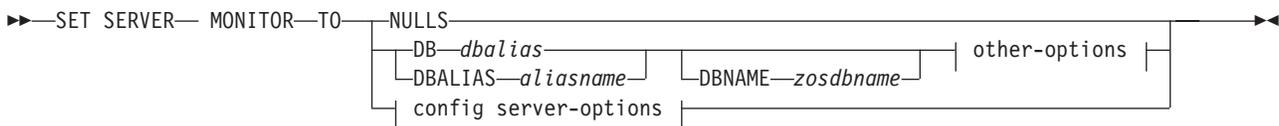
```
SET OUTPUT MONITOR SCRIPT "monitor.sql"
```

SET SERVER command

Use the SET SERVER command to specify the database that is used as a monitor control server 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.

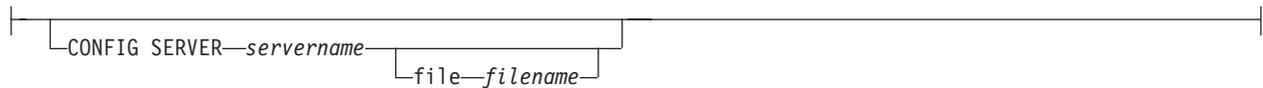
Syntax



other-options:



config server-options:



Parameters

MONITOR

Specify to set the database as a monitor control 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 of a z/OS subsystem or Linux, UNIX, Windows, or System i database as cataloged on the DB2 from which the ASNCLP is invoked. This keyword is deprecated.

DBALIAS *aliasname*

Specifies the database alias name of a z/OS subsystem or Linux, UNIX, Windows, or System i database as cataloged on the DB2 from which the ASNCLP is invoked.

DBNAME *zosdbname*

z/OS Specifies the z/OS database name. This is a logical z/OS database name, as created on a z/OS subsystem.

other-options clause:

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.

config server-options clause:

CONFIG SERVER *servername*

UNIX System Services (USS) on z/OS: Specifies the DB2 database to use as a monitor control server when the ASNCLP program is running on USS. The server name must match the bracketed [NAME] field that is entered in the ASNCLP configuration file.

FILE *filename*

Specifies the complete path and file name to the ASNCLP configuration file. If you do not use the FILE parameter, the ASNCLP program attempts to use the asnservers.ini file in the current directory, if that file exists.

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 (PERFORMANCE):

```
SUBSTITUTE CONTACT REPLADMIN WITH PERFORMANCE
```

Product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product interface, in a suite-wide information center, and in PDF file books.

You can also order IBM® publications in hardcopy format online or through your local IBM representative.

To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order.

You can send your comments about documentation in the following ways:

- Online reader comment form: www.ibm.com/software/data/rcf/
- E-mail: comments@us.ibm.com

Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You can also provide feedback on products and documentation.

Customer support

For customer support for IBM products and for product download information, go to the support and downloads site at www.ibm.com/support/us/.

You can open a support request by going to the software support service request site at www.ibm.com/software/support/probsub.html.

My IBM

You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at www.ibm.com/account/us/.

Software services

For information about software, IT, and business consulting services, go to the solutions site at www.ibm.com/businesssolutions/us/en.

Information Management product support

For Information Management product support, news, and other product information, go to the Information Management support site at www.ibm.com/software/data/support/.

Federation, replication, and event publishing products support

For support, go to:

- IBM InfoSphere Federation Server
www.ibm.com/software/data/integration/support/federation_server/

- IBM InfoSphere Replication Server
www.ibm.com/software/data/integration/support/replication_server/
- IBM InfoSphere Data Event Publisher
www.ibm.com/software/data/integration/support/data_event_publisher/

Classic products support

For support, go to:

- IBM InfoSphere Classic Federation Server for z/OS
www.ibm.com/software/data/integration/support/classic_federation_server_z/
- IBM InfoSphere Classic Replication Server for z/OS
www.ibm.com/software/data/infosphere/support/replication-server-z/
- IBM InfoSphere Classic Data Event Publisher for z/OS
www.ibm.com/software/data/integration/support/data_event_publisher_z/
- IBM InfoSphere Data Integration Classic Connector for z/OS
www.ibm.com/software/data/integration/support/data_integration_classic_connector_z/

General information

To find general information about IBM, go to www.ibm.com.

Product feedback

You can provide general product feedback through the Consumability Survey at www.ibm.com/software/data/info/consumability-survey.

Documentation feedback

You can click the feedback link in any topic in the information center to comment on the information center.

You can also send your comments about PDF file books, the information center, or any other documentation in the following ways:

- Online reader comment form: www.ibm.com/software/data/rcf/
- E-mail: comments@us.ibm.com

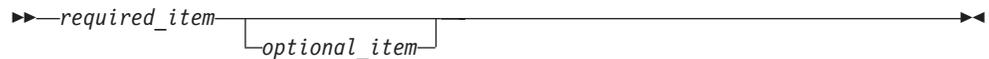
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
 - The >>--- symbol indicates the beginning of a syntax diagram.
 - The ---> symbol indicates that the syntax diagram is continued on the next line.
 - The >--- symbol indicates that a syntax diagram is continued from the previous line.
 - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).



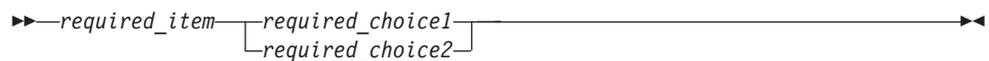
- Optional items appear below the main path.



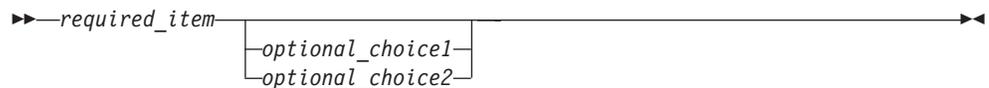
If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.



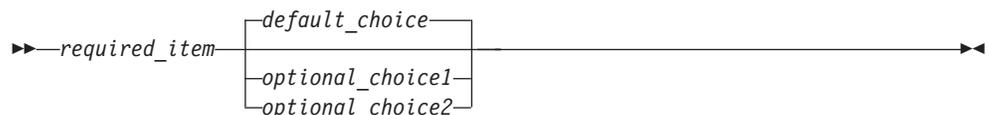
- If you can choose from two or more items, they appear vertically, in a stack. If you must choose one of the items, one item of the stack appears on the main path.



If choosing one of the items is optional, the entire stack appears below the main path.



If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.



If the repeat arrow contains a comma, you must separate repeated items with a comma.



A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



Fragment-name:



- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown.
- Variables appear in all lowercase italic letters (for example, *column-name*). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).

Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere™ Information Server product modules and user interfaces are not fully accessible. The installation program installs the following product modules and components:

- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- InfoSphere DataStage® and QualityStage Administrator
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench

For more information about IBM product accessibility status, go to http://www.ibm.com/able/product_accessibility/index.html.

Accessible documentation

Accessible documentation for IBM Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.

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 character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample 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 on their first occurrence in this information with the appropriate symbol.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The following terms are trademarks or registered trademarks of other companies:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds 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.

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.

UNIX is a registered trademark of The Open Group in the United States and 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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both

Other company, product, or service names may be trademarks or service marks of others.

Index

A

accessibility 285
ALTER ADD COLUMN command (Q replication) 86, 211
ALTER ALERT CONDITIONS FOR APPLY command 251
ALTER ALERT CONDITIONS FOR CAPTURE command 253
ALTER ALERT CONDITIONS FOR QAPPLY command 256
ALTER ALERT CONDITIONS FOR QCAPTURE command 258
ALTER CAPTURE PARAMETERS command (Q replication) 88
ALTER CONFIGURATION APPLY command 88
ALTER CONTACT command 260
ALTER GROUP command 261
ALTER MEMBER ADD COLS command (SQL replication) 11
ALTER MONITOR SUSPENSION command 261
ALTER MONITOR SUSPENSION TEMPLATE command 263
ALTER PUB command (event publishing) 212
ALTER PUBQMAP command (event publishing) 214
ALTER QSUB command (bidirectional replication) 154
ALTER QSUB command (peer-to-peer replication) 157
ALTER QSUB command (unidirectional Q replication) 91
ALTER REGISTRATION command (SQL replication) 12
ALTER REPLQMAP command (Q replication) 89, 152
ALTER SUBSCRIPTION SET command (SQL replication) 15
ASNCLP
 configuration file 4
 PROMOTE PUB command 232
 PROMOTE PUBQMAP 234
 SET PROFILE (SQL replication) 49
ASNCLP commands 127, 182
 ALTER ADD COLUMN (Q replication) 86, 211
 ALTER ALERT CONDITIONS FOR APPLY 251
 ALTER ALERT CONDITIONS FOR CAPTURE 253
 ALTER ALERT CONDITIONS FOR QAPPLY 256
 ALTER ALERT CONDITIONS FOR QCAPTURE 258
 ALTER CAPTURE PARAMETERS (Classic replication) 88
 ALTER CONFIGURATION APPLY 88
 ALTER CONTACT 260

ASNCLP commands (*continued*)
 ALTER GROUP 261
 ALTER MEMBER ADD COLS (SQL replication) 11
 ALTER MONITOR SUSPENSION 261
 ALTER MONITOR SUSPENSION TEMPLATE 263
 ALTER PUB (event publishing) 212
 ALTER PUB command (event publishing) 212
 ALTER PUBQMAP (event publishing) 214
 ALTER QSUB (bidirectional replication) 154
 ALTER QSUB (peer-to-peer replication) 157
 ALTER QSUB (unidirectional Q replication) 91
 ALTER REGISTRATION (SQL replication) 12
 ALTER REPLQMAP (Q replication) 89, 152
 ALTER SUBSCRIPTION SET (SQL replication) 15
ASNCLP SESSION SET TO (Q replication) 95, 158, 215
ASNCLP SESSION SET TO (SQL replication) 17
CREATE ALERT CONDITIONS FOR APPLY 264
CREATE ALERT CONDITIONS FOR CAPTURE 266
CREATE ALERT CONDITIONS FOR QAPPLY 268
CREATE ALERT CONDITIONS FOR QCAPTURE 269
CREATE CONTACT 271
CREATE CONTROL TABLES (SQL replication) 272
CREATE CONTROL TABLES FOR (Q replication) 96, 159, 216
CREATE CONTROL TABLES FOR (SQL replication) 18
CREATE GROUP 274
CREATE MEMBER (SQL replication) 20
CREATE MONITOR SUSPENSION 274
CREATE MONITOR SUSPENSION TEMPLATE 276
CREATE PUB command (event publishing) 222
CREATE PUBQMAP (event publishing) 226
CREATE QSUB (bidirectional replication) 167
CREATE QSUB (peer-to-peer replication) 171
CREATE QSUB (unidirectional Q replication) 103

ASNCLP commands (*continued*)
 CREATE REGISTRATION (SQL replication) 28
 CREATE REPLQMAP (Q replication) 102, 165
 CREATE SUBSCRIPTION SET (SQL replication) 35
 DELEGATE CONTACT 277
 DROP ALERT CONDITIONS FOR APPLY 277
 DROP ALERT CONDITIONS FOR CAPTURE 278
 DROP ALERT CONDITIONS FOR QAPPLY 278
 DROP ALERT CONDITIONS FOR QCAPTURE 278
 DROP CONTACT 279
 DROP CONTROL TABLES (Q replication) 119, 174, 228
 DROP CONTROL TABLES (SQL replication) 37
 DROP GROUP 279
 DROP MEMBER (SQL replication) 38
 DROP MONITOR SUSPENSION 280
 DROP MONITOR SUSPENSION TEMPLATE 280
 DROP PUB (event publishing) 228
 DROP PUB command (event publishing) 228
 DROP PUBQMAP (event publishing) 229
 DROP QSUB (unidirectional Q replication) 120
 DROP REGISTRATION (SQL replication) 39
 DROP REPLQMAP (Q replication) 119, 175
 DROP STMT (SQL replication) 39
 DROP SUBGROUP (multidirectional Q replication) 175
 DROP SUBSCRIPTION SET (SQL replication) 40
 DROP SUBTYPE (bidirectional replication) 176
 DROP SUBTYPE (peer-to-peer replication) 176
 interactive mode, using 3
 LIST APPLY SCHEMA (Q replication and event publishing) 122, 123, 177, 178, 229
 LIST CAPTURE SCHEMA (Q replication and event publishing) 123, 178, 229
 LIST MONITOR SUSPENSION 281
 LIST MONITOR SUSPENSION TEMPLATE 281
 LIST PUBQMAPS 231
 LIST PUBS 230
 LIST QSUB (Q replication) 121
 LIST REPLQMAP (Q replication) 124

ASNCLP commands (*continued*)

- LIST SCHEMA (Q replication and event publishing) 123, 178, 229
- LOAD DONE (Q replication) 126, 179
- LOAD MULTIDIR REPL SCRIPT command (multidirectional Q replication) 181
- multidirectional Q replication 151
- OFFLINE LOAD (SQL replication) 41
- PROMOTE REGISTRATION (SQL replication) 42
- PROMOTE REPLQMAP 129, 183
- PROMOTE SUBSCRIPTION SET (SQL replication) 44
- Replication Alert Monitor 247
- SET APPLY SCHEMA (Q replication) 130, 185
- SET BIDI NODE 185
- SET BIDIRECTIONAL NODE 185
- SET CAPTURE SCHEMA (Q replication) 131, 187, 235
- SET CAPTURE SCHEMA (SQL replication) 45
- SET CONNECTION (multidirectional Q replication) 188
- SET DROP (SQL replication) 46
- SET DROP (unidirectional replication) 131
- SET ENFORCE MATCHING CONSTRAINTS (multidirectional Q replication) 189
- SET LOG (Q replication) 132, 189, 236
- SET LOG (SQL replication) 47
- SET MULTIDIR SCHEMA (multidirectional Q replication) 190
- SET OUTPUT (monitor) 281
- SET OUTPUT (SQL replication) 48
- SET OUTPUT (unidirectional Q replication) 133, 236
- SET OUTPUT command (multidirectional Q replication) 190
- SET PEER NODE 191
- SET PEERTOPEER NODE 191
- SET PROFILE (Q replication) 133, 193
- SET QMANAGER (Q replication) 137, 196, 237
- SET REFERENCE TABLE command (multidirectional Q replication) 197
- SET RUN SCRIPT (Q replication) 138, 238
- SET RUN SCRIPT (SQL replication) 52
- SET SERVER (multidirectional Q replication) 198
- SET SERVER (Q replication and event publishing) 141, 241
- SET SERVER (Replication Alert Monitor) 282
- SET SERVER (SQL replication) 55
- SET SUBGROUP (multidirectional Q replication) 200
- SET TABLES (multidirectional Q replication) 200

ASNCLP commands (*continued*)

- SET TRACE (Q replication) 144, 201, 243
- SET TRACE (SQL replication) 58
- SHOW SET ENV (Q replication) 144, 202
- SQL replication 7
- START PUB (event publishing) 222, 244
- START PUB command (event publishing) 244
- START QSUB (Q replication) 144, 202
- STOP PUB (event publishing) 244
- STOP PUB command (event publishing) 244
- STOP QSUB (Q replication) 146, 203
- SUBSTITUTE CONTACT 283
- unidirectional Q replication 85
- using input file 4
- VALIDATE WSMQ ENVIRONMENT FOR 147, 205, 245
- VALIDATE WSMQ MESSAGE FLOW FOR REPLQMAP 148, 206

ASNCLP program

- getting started 1

ASNCLP scripts, using 4

ASNCLP SESSION SET TO command (Q replication) 95, 158, 215

ASNCLP SESSION SET TO command (SQL replication) 17

B

- batch mode, using 4
- bidirectional Q replication
 - sample scripts 65
- binding 2

C

Classic replication

- configuration file 4
- CLASSPATH environment variable 1
- CREATE ALERT CONDITIONS FOR APPLY command 264
- CREATE ALERT CONDITIONS FOR CAPTURE command 266
- CREATE ALERT CONDITIONS FOR QAPPLY command 268
- CREATE ALERT CONDITIONS FOR QCAPTURE command 269
- CREATE CONTACT command 271
- CREATE CONTROL TABLES command (SQL replication) 272
- CREATE CONTROL TABLES FOR command (Q replication) 96, 159, 216
- CREATE CONTROL TABLES FOR command (SQL replication) 18
- CREATE GROUP command 274
- CREATE MEMBER command (SQL replication) 20
- CREATE MONITOR SUSPENSION command 274
- CREATE MONITOR SUSPENSION TEMPLATE command 276

- CREATE PUB command (event publishing) 222
- CREATE PUBQMAP command (event publishing) 226
- CREATE QSUB command (bidirectional replication) 167
- CREATE QSUB command (peer-to-peer replication) 171
- CREATE QSUB command (unidirectional Q replication) 103
- CREATE REGISTRATION command (SQL replication) 28
- CREATE REPLQMAP command (Q replication) 102, 165
- CREATE STMT command (SQL replication)
 - ASNCLP commands 33
 - CREATE STMT (SQL replication) 33
- CREATE SUBSCRIPTION SET command (SQL replication) 35
- customer support 285

D

- DELEGATE CONTACT command 277
- documentation
 - accessible 285
- DROP ALERT CONDITIONS FOR APPLY command 277
- DROP ALERT CONDITIONS FOR CAPTURE command 278
- DROP ALERT CONDITIONS FOR QAPPLY command 278
- DROP ALERT CONDITIONS FOR QCAPTURE command 278
- DROP CONTACT command 279
- DROP CONTROL TABLES command (SQL replication) 37
- DROP CONTROL TABLES ON command (Q replication) 119, 174, 228
- DROP GROUP command 279
- DROP MEMBER command (SQL replication) 38
- DROP MONITOR SUSPENSION command 280
- DROP MONITOR SUSPENSION TEMPLATE command 280
- DROP PUB command (event publishing) 228
- DROP PUBQMAP command (event publishing) 229
- DROP QSUB command (unidirectional Q replication) 120
- DROP REGISTRATION command (SQL replication) 39
- DROP REPLQMAP command (Q replication) 119, 175
- DROP STMT command (SQL replication) 39
- DROP SUBGROUP command (multidirectional Q replication) 175
- DROP SUBSCRIPTION SET command (SQL replication) 40
- DROP SUBTYPE command (bidirectional replication) 176
- DROP SUBTYPE command (peer-to-peer replication) 176

E

event publishing
sample scripts 207, 208

I

IBM support 285
input file for commands
ASNCLP commands 4
interactive mode 3
introduction, ASNCLP program 1

J

Java environment, setting up 1

L

legal notices 291
LIST APPLY SCHEMA command 122,
123, 177, 178, 229
LIST CAPTURE SCHEMA
command 123, 178, 229
LIST MONITOR SUSPENSION
command 281
LIST MONITOR SUSPENSION
TEMPLATE command 281
LIST PUBQMAPS command 231
LIST PUBS command 230
LIST QSUB command (Q
replication) 121
LIST REPLQMAP command (Q
replication) 124
LIST SCHEMA command 123, 178, 229
LOAD DONE command (Q
replication) 126, 179
LOAD MULTIDIR REPL SCRIPT
command (multidirectional Q
replication) 181

M

multidirectional Q replication
commands 151

O

OFFLINE LOAD command (SQL
replication) 41
operating systems, supported 1
Oracle replication
configuration file 4

P

peer-to-peer configurations
promoting 82
peer-to-peer Q replication (three servers)
sample scripts 75
peer-to-peer Q replication (two servers)
sample scripts 70
product accessibility
accessibility 289
PROMOTE PUB command 232

PROMOTE PUBQMAP command 234
PROMOTE QSUB 127, 182
PROMOTE QSUB command 127, 182
PROMOTE REGISTRATION command
(SQL replication) 42
PROMOTE REPLQMAP command 129,
183
PROMOTE SUBSCRIPTION SET
command (SQL replication) 44
promoting
publications 232
Promoting
publishing queue maps 234
promoting peer-to-peer configurations
sample scripts 82
promoting subscriptions 127, 182
promoting unidirectional
configurations 81

Q

Q subscriptions
ALTER QSUB 91
changing 91

R

Replication Alert Monitor
ASNCLP commands 247
sample scripts 248

S

sample scripts
bidirectional Q replication 65
event publishing 207, 208
peer-to-peer Q replication (three
servers) 75
peer-to-peer Q replication (two
servers) 70
promoting peer-to-peer
configurations 82
Replication Alert Monitor 248
SQL replication 8
unidirectional Q replication 59, 63
screen readers 285
SET APPLY SCHEMA command (Q
replication) 130, 185
SET BIDI NODE command 185
SET BIDIRECTIONAL NODE
command 185
SET CAPTURE SCHEMA command (Q
replication) 131, 187, 235
SET CAPTURE SCHEMA command (SQL
replication) 45
SET CONNECTION command
(multidirectional Q replication) 188
SET DROP command (SQL
replication) 46
SET DROP command (unidirectional
replication) 131
SET ENFORCE MATCHING
CONSTRAINTS command
(multidirectional Q replication) 189
SET LOG command (Q replication) 132,
189, 236

SET LOG command (SQL replication) 47
SET MULTIDIR SCHEMA command
(multidirectional Q replication) 190
SET OUTPUT command (monitor) 281
SET OUTPUT command (multidirectional
Q replication) 190
SET OUTPUT command (SQL
replication) 48
SET OUTPUT command (unidirectional
Q replication) 133, 236
SET PEER NODE command 191
SET PEERTOPEER NODE
command 191
SET PROFILE command (Q
replication) 133, 193
SET PROFILE command (SQL
replication) 49
SET QMANAGER command (Q
replication) 137, 196, 237
SET REFERENCE TABLE command
(multidirectional Q replication) 197
SET RUN SCRIPT command (Q
replication) 138, 238
SET RUN SCRIPT command (SQL
replication) 52
SET SERVER command (multidirectional
Q replication) 198
SET SERVER command (Q replication
and event publishing) 141, 241
SET SERVER command (Replication Alert
Monitor) 282
SET SERVER command (SQL
replication) 55
SET SUBGROUP command
(multidirectional Q replication) 200
SET TABLES command (multidirectional
Q replication) 200
SET TRACE command (Q
replication) 144, 201, 243
SET TRACE command (SQL
replication) 58
SHOW SET ENV command (Q
replication) 144, 202
software services 285
SQL replication
ASNCLP commands 7
sample scripts 8
START PUB command (event
publishing) 244
START QSUB command (Q
replication) 144, 202
STOP PUB command (event
publishing) 244
STOP QSUB command (Q
replication) 146, 203
subscriptions
promoting 127, 182
SUBSTITUTE CONTACT command 283
support, customer 285
supported operating systems 1

T

trademarks 293

U

unidirectional Q replication
 commands 85
 sample scripts 59, 63

V

VALIDATE WSMQ ENVIRONMENT
 FOR command 147, 205, 245
VALIDATE WSMQ MESSAGE FLOW
 FOR REPLQMAP command 148, 206

Z

z/OS packages, binding 2



Printed in USA

SC19-1018-04



Spine information:

IBM InfoSphere Replication Server IBM InfoSphere Data Event Publisher

Version 9.7

ASNCLP Program Reference for Replication and Event Publishing

