IBM DB2 Information Integrator

ASNCLP Program Reference for Replication and Event Publishing

Version 8.2

SC18-9410-00

IBM DB2 Information Integrator

ASNCLP Program Reference for Replication and Event Publishing

Version 8.2

SC18-9410-00

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 215.

You can order IBM publications online or through your local IBM representative:

- To order publications online, go to the IBM Publications Center at www.ibm.com/shop/publications/order
- To find your local IBM representative, go to the IBM Directory of Worldwide Contacts at www.ibm.com/planetwide

Contents

Chapter 7. Offline load utility commands for SQL replication 69

Part 3. Commands for Q replication and event publishing 71

Chapter 8. Environment commands for unidirectional Q replication 73

Chapter 9. Environment commands for bidirectional and peer-to-peer Q replication 83

Chapter 10. Control table definition commands for Q replication 89

Appendix B. Dotted decimal diagrams 173

DB2 Information Integrator documentation 203

Accessibility 213

Notices 215

Index 219

Contacting IBM 227

About this document

DB2® Information Integrator replication provides three interfaces for setting up and maintaining replication environments:

- The DB2 Information Integrator Replication Center
- The ASNCLP program
- OS/400®-system commands (for iSeries™ platform only)

For information about the DB2 Replication Center or the OS/400-system commands for replication, see the *DB2 Information Integrator SQL Replication Guide and Reference Version 8.2* (SC27-1121) or the replication information in the DB2 Information Center. For information about Q replication and event publishing tasks, see *DB2 Information Integrator Replication and Event Publishing Guide and Reference* (SC18-7568).

For information on known issues and troubleshooting, see Appendix A, “Frequently asked questions,” on page 167.

Who should read this document

This document is primarily for database administrators and system administrators who must administer and maintain a data replication or event publishing environment. You should be very familiar with standard database terminology, have a working knowledge of the operating systems that are involved in replication, and have experience with database design, database administration, database performance analysis, server connectivity, and networking. You should understand the applications in your environment and how they manipulate the data that you want to replicate. You should be familiar with replication concepts and components.

This document is a companion document to the *DB2 Information Integrator SQL Replication Guide and Reference Version 8.2* (SC27-1121), the replication information in the DB2 Information Center, and the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* (SC18-7568).

Conventions

This document uses the following highlighting conventions:

- **Boldface type** indicates commands or user interface controls such as names of fields, folders, icons, or menu choices.
- `Monospace type` indicates examples of text that you enter exactly as shown.
- *Italic type* indicates variables that you should replace with a value. It is also used to indicate book titles and for emphasis of words.

How to read syntax diagrams

The following rules apply to the syntax diagrams used in this book:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line.

  The ▶▶── symbol indicates the beginning of a statement.

The ───► symbol indicates that the statement syntax is continued on the next line.

The ►─── symbol indicates that a statement is continued from the previous line.

The ───►◄ symbol indicates the end of a statement.

Diagrams of syntactical units other than complete statements start with the ►─── symbol and end with the ───► symbol.

- Keywords, their allowable synonyms, and reserved parameters, are either shown in uppercase or lowercase, depending on the operating system. These items must be entered exactly as shown. Variables appear in lowercase italics (for example, *column-name*). They represent user-defined parameters or suboptions.

  When entering commands, separate the parameters and keywords by at least one space if there is no intervening punctuation.
- Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs, and so on) and numbers exactly as given.
- Footnotes are shown by a number in parentheses, for example, (1).
- Required items appear on the horizontal line (the main path).

```
►►──required_item──────────────────────────────────────────────────►◄
```

- A parameter's default value is displayed above the path:

```
                    ┌─default_value─┐
►►──required_item───┴───────────────┴──────────────────────────────►◄
```

- Optional items appear below the main path.

```
►►──required_item───┬───────────────┬──────────────────────────────►◄
                    └─optional_item─┘
```

- 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.

```
►►──required_item───┬─required_choice1─┬───────────────────────────►◄
                    └─required_choice2─┘
```

  If choosing one of the items is optional, the entire stack appears below the main path.

```
►►──required_item───┬──────────────────┬───────────────────────────►◄
                    ├─optional_choice1─┤
                    └─optional_choice2─┘
```

If you can choose from two or more items, the value at the top of the stack is the default.

Part 1. General information

Chapter 1. Introduction

This document describes the ASNCLP program. The ASNCLP program is a command-line interface for administration of SQL replication, Q replication, and event publishing.

The ASNCLP program runs on Linux, UNIX®, or Windows® operating systems. It will also work on catalogued z/OS® and iSeries databases.

This chapter presents an overview of the ASNCLP program and contains the following topics:

- "Comparison of the ASNCLP program with other replication interfaces"
- "Comparison of ASNCLP commands for SQL replication with commands for Q replication and event publishing"
- "Operating-system environments that are supported" on page 4
- "The ASNCLP commands" on page 4

Comparison of ASNCLP commands for SQL replication with commands for Q replication and event publishing

Some of the ASNCLP commands for SQL replication are similar or identical to those for Q replication and event publishing. For example, many of the same environment commands are used. However, most commands are very different for SQL replication than for Q replication, as the program and control table architecture for the two types of replication are very different.

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for more information on different types of replication.

Comparison of the ASNCLP program with other replication interfaces

DB2 II replication provides three interfaces for administration. The first is the Replication Center, a graphical user interface. The other two are command-line interfaces. The primary command-line interface is ASNCLP, which provides administration for all replication configurations on all supported platforms. The other command-line interface, which is used with native iSeries commands, can only be used with SQL replication on the iSeries platform.

In previous versions of DB2 replication, such as DB2 DataPropagator™ V7 or DB2 DataJoiner® V2, another graphical user interface was provided, DataJoiner Replication Administration (DJRA). It is not supported with Version 8 of DB2 replication and event publishing

DB2 DataPropagator for iSeries provides OS/400-native SQL replication administration commands for Version 8.2. These commands are similar to those provided by DB2 DataPropagator for AS/400® Version 5 and DataPropagator Relational/400 Version 1. The native OS/400 commands support the iSeries platform in the following manner:

- Registration tasks work only for an iSeries source.

- Subscription tasks (subscription set and subscription-set member) always assume that the logical replication servers (Capture control server, Apply control server, and target server) all reside on iSeries servers.

The existing OS/400 naming convention for these commands is specific to the OS/400 environment and is different from the ASNCLP naming convention. The function that is provided by both sets of commands is similar, except for some OS/400-specific parameters.

For DB2 Version 7, the DJRA tool provided a set of administration commands to set up DB2 replication. These commands are not supported in DB2 Information Integrator Version 8.2 and have been replaced by the ASNCLP program.

All of the DJRA commands are supported by the ASNCLP program, except the commands for **promote table** and **promote table space**. The **promote table** and **promote table space** commands are not replication specific, and you can use the DB2 Control Center or command-line processor to do these tasks.

Operating-system environments that are supported

The ASNCLP administrative commands runs on the Windows, Linux, and UNIX operating-system environments only. The ASNCLP commands will not run natively in the z/OS or OS/400 environments.

However, the ASNCLP commands will generate replication definitions for all operating-system environments that are supported by the replication products: for DB2 Information Integrator Version 8.2, z/OS, OS/390®, OS/400 (SQL replication only), UNIX (AIX®, Solaris Operating Environment, HP-UX), Linux, and Microsoft Windows. You must have connectivity to each server for which you are generating replication definitions; that is, you must be able to issue a **db2 connect** statement to each of the other servers.

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

The ASNCLP commands

The ASNCLP program has two general types of commands: environment and task. The environment commands define settings that are used by task commands. An example of an environment command is a setting that you can use to identify a source server to be the default for all task commands. Task commands are used to perform replication tasks such as creating a subscription.

Environment commands have a scope. The scope is from the time the command is issued to one of the following events occurs:

- You end the command-line session if you are in an interactive session.
- The ASNCLP program reaches the end of the input file if you are not in an interactive session.
- You change the value that environment command set by issuing a new environment command.

ASNCLP commands for SQL replication

The following table lists the actions for setting up your replication environment and where you can find descriptions of the ASNCLP program SQL task commands to perform those actions.

Table 1. Commands for setting up replication environment (SQL replication)

Action	Command descriptions
Create control tables	"CREATE CONTROL TABLES command" on page 35
Drop control tables	"DROP CONTROL TABLES command" on page 38
Create registration	"CREATE REGISTRATION command" on page 41
Alter registration	"ALTER REGISTRATION command" on page 45
Drop registration	"DROP REGISTRATION command" on page 48
Promote registration	"PROMOTE REGISTRATION command" on page 48
Create subscription set	"CREATE SUBSCRIPTION SET command" on page 51
Alter subscription set	"ALTER SUBSCRIPTION SET command" on page 53
Drop subscription set	"DROP SUBSCRIPTION SET command" on page 65
Promote subscription set	"PROMOTE SUBSCRIPTION SET command" on page 65
Create member	"CREATE MEMBER command" on page 54
Add columns to an existing member	"ALTER MEMBER ADD COLS command" on page 62
Drop member	"DROP MEMBER command" on page 61
Create a statement for an existing subscription set	"CREATE STMT command" on page 63
Drop statements for an existing subscription set	"DROP STMT command" on page 64
Control a manual full refresh for offline load procedures	"OFFLINE LOAD command" on page 69
Establish a session for your chosen type of replication	"SET SESSION command" on page 24
Assigns a database alias for a logical replication server or specify a user ID and password to use when connecting to the database	"SET SERVER command" on page 25
Set up customization rules for creating table space objects	"SET PROFILE command" on page 27
Determine whether to drop the table space when you drop the database object that it contains	"SET DROP command" on page 29
Define output files for the replication command line interface	"SET OUTPUT command" on page 31
Define the log file for the replication command line interface	"SET LOG command" on page 31
Set a default source and target Capture schema for all task commands	"SET CAPTURE SCHEMA command" on page 32
Enable and disable the trace for the ASNCLP commands	"SET TRACE command" on page 32
Specify whether to automatically run each task command from an input file before the ASNCLP commands process the next task command	"SET RUN SCRIPT command" on page 33

ASNCLP commands for Q replication

The following table lists the actions for unidirectional replication and where you can find descriptions of the ASNCLP commands to perform those actions.

Table 2. Commands for unidirectional replication (Q replication)

Action	Command descriptions	Sample files
Establish a session for your chosen type of replication	“SET SESSION command” on page 73	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the server (database) used in the ASNCLP session and authentication information and other required parameters for connecting to the server	“SET SERVER command” on page 74	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the WebSphere® queue manager	“SET QMANAGER command” on page 74	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify custom parameters for database objects to be created implicitly	“SET PROFILE command” on page 75	CrtQApplyZOS.txt, CrtQApplyZOSBuffer.txt, CrtQCaptureZOS.txt, and CrtQSubsZOS.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Determine whether to drop the table space when you drop the database object that it contains	“SET DROP command” on page 78	Not available
Define output files for the replication command line interface	“SET OUTPUT command” on page 78	CrtQApplyZOSBuffer.t and xt CrtQSubZOSQPro.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Define the log file for the replication command line interface	“SET LOG command” on page 80	Not available
Enable and disable the trace for the ASNCLP commands	“SET TRACE command” on page 80	Not available
Specify whether to automatically run each task command from an input file before the ASNCLP commands process the next task command	“SET RUN SCRIPT command” on page 80	CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set a default source and target Q Capture schema for all task commands	“SET CAPTURE SCHEMA command” on page 81	CrtQCaptureUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set a default source and target Q Apply schema for all task commands	“SET APPLY SCHEMA command” on page 82	CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Display the environment set during the session	“SHOW SET ENV command” on page 82	Not available
Create the control tables for the Q Capture and Q Apply programs	“CREATE CONTROL TABLES command” on page 89	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Drop the control tables for the Q Capture and Q Apply programs	“DROP CONTROL TABLES command” on page 94	DropQCaptureUOW.txt and DropQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)

Table 2. Commands for unidirectional replication (Q replication) (continued)

Action	Command descriptions	Sample files
Create a replication queue map	"CREATE REPLQMAP command" on page 101	CrtReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a replication queue map	"ALTER REPLQMAP command" on page 102	AlterQReplMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a replication queue map	"DROP REPLQMAP command" on page 104	DropReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create a Q subscription	"CREATE QSUB command (unidirectional replication)" on page 113	CrtQSubsUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a Q subscription	"ALTER QSUB command" on page 119	AlterQSubsUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a Q subscription	"DROP QSUB command" on page 121	DropQSubs.txt (located in the /sqllib/samples/repl/asnclp/ directory)

The following table lists the actions for bidirectional replication and where you can find descriptions of the ASNCLP commands to perform those actions.

Table 3. Commands for bidirectional replication (Q replication)

Action	Command descriptions	Sample files
Set the Q Capture and Q Apply schema to the value specified on the specified server for bidirectional or peer-to-peer replication processing	"SET MULTIDIR SCHEMA command" on page 84	BiDirectionalInput.txt, DropBiDirectionalInput.txt, DropSubGBiInput.txt, and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the subgroup name of the bidirectional or peer-to-peer replication scenario	"SET SUBGROUP command" on page 84	BiDirectionalInput.txt, DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, DropSubGBiInput.txt, P2PQFactoryInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the connection for the direction between the nodes	"SET CONNECTION command" on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the source and target tables that participate in the bidirectional or peer-to-peer replication setup	"SET TABLES command" on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)

Table 3. Commands for bidirectional replication (Q replication) (continued)

Action	Command descriptions	Sample files
Set the reference table for the Q subscription	"SET REFERENCE TABLE command" on page 86	DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create the control tables for the Q Capture and Q Apply programs on both participating servers	"CREATE CONTROL TABLES command" on page 89	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory).
Drop the control tables for the Q Capture and Q Apply programs	"DROP CONTROL TABLES command" on page 94	DropQCaptureUOW.txt and DropQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create two replication queue maps, one for replicating in each direction of your bidirectional configuration	"CREATE REPLQMAP command" on page 101	CrtReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a replication queue map	"ALTER REPLQMAP command" on page 102	AlterQReplMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a replication queue map	"DROP REPLQMAP command" on page 104	DropReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create a bidirectional replication scenario	"CREATE QSUB command (bidirectional replication)" on page 125	Sample files in the folder bidir
Alter a bidirectional Q subscription	"ALTER QSUB command (bidirectional replication)" on page 127	Update3.in (located in the /sqllib/samples/repl/asnclp/ update directory)
Delete a Q subscription	"DROP QSUB command" on page 121	DropQSubs.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a subgroup	"DROP SUBGROUP command" on page 87	DropSubGBiInput.txt (located in the /sqllib/samples/repl/asnclp/ delete directory)

The following table lists the actions for peer-to-peer replication between two servers and where you can find descriptions of the ASNCLP commands to perform those actions.

Table 4. Commands for peer-to-peer replication between two servers (Q replication)

Action	Command descriptions	Sample files
Set the Q Capture and Q Apply schema to the value specified on the specified server for bidirectional or peer-to-peer replication processing	"SET MULTIDIR SCHEMA command" on page 84	BiDirectionalInput.txt, DropBiDirectionalInput.txt, DropSubGBiInput.txt, and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the subgroup name of the bidirectional or peer-to-peer replication scenario	"SET SUBGROUP command" on page 84	BiDirectionalInput.txt, DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, DropSubGBiInput.txt, P2PQFactoryInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the connection for the direction between the nodes	"SET CONNECTION command" on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the source and target tables that participate in the bidirectional or peer-to-peer replication setup	"SET TABLES command" on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the reference table for the Q subscription	"SET REFERENCE TABLE command" on page 86	DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create the control tables for the Q Capture and Q Apply programs on both participating nodes	"CREATE CONTROL TABLES command" on page 89	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create two replication queue maps, one for replicating in each direction of your peer-to-peer configuration	"CREATE REPLQMAP command" on page 101	CrtReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a replication queue map	"ALTER REPLQMAP command" on page 102	AlterQReplMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a replication queue map	"DROP REPLQMAP command" on page 104	DropReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create peer-to-peer Q subscriptions	"CREATE QSUB command (peer-to-peer replication)" on page 131	Sample files in the folder 2nodes (located in the /sqllib/samples/repl/asnclp/ directory)

Table 4. Commands for peer-to-peer replication between two servers (Q replication) (continued)

Action	Command descriptions	Sample files
Alter a peer-to-peer Q subscription	“ALTER QSUB command (peer-to-peer replication)” on page 132	Not available
Delete a peer-to-peer Q subscription	“DROP SUBTYPE command (peer-to-peer replication)” on page 134	delete2.in (located in the /sqllib/samples/repl/asnclp/ delete directory)
Delete a subgroup	“DROP SUBGROUP command” on page 87	DropSubGBiInput.txt (located in the /sqllib/samples/repl/asnclp/ delete directory)

The following table lists the actions for peer-to-peer replication between three or more servers and where you can find descriptions of the ASNCLP commands to perform those actions.

Table 5. Commands for peer-to-peer replication with three or more servers (Q replication)

Action	Command descriptions	Sample files
Set the Q Capture and Q Apply schema to the value specified on the specified server for bidirectional or peer-to-peer replication processing	“SET MULTIDIR SCHEMA command” on page 84	BiDirectionalInput.txt, DropBiDirectionalInput.txt, DropSubGBiInput.txt, and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the subgroup name of the bidirectional or peer-to-peer replication scenario	“SET SUBGROUP command” on page 84	BiDirectionalInput.txt, DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, DropSubGBiInput.txt, P2PQFactoryInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the connection for the direction between the nodes	“SET CONNECTION command” on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Specify the source and target tables that participate in the bidirectional or peer-to-peer replication setup	“SET TABLES command” on page 85	BiDirectionalInput.txt and P2PQFactoryInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Set the reference table for the Q subscription	“SET REFERENCE TABLE command” on page 86	DropBIInput.txt, DropBiDirectionalInput.txt, DropLogTblInput.txt, UpdateBiInput.txt, and UpdateLogTblInput.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create the control tables for the Q Capture and Q Apply programs on all participating DB2 UDB nodes	“CREATE CONTROL TABLES command” on page 89	CrtQCaptureUOW.txt and CrtQApplyUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)

Table 5. Commands for peer-to-peer replication with three or more servers (Q replication) (continued)

Action	Command descriptions	Sample files
Create two replication queue maps for every pair of nodes, one for replicating in each direction between the nodes in a pair	"CREATE REPLQMAP command" on page 101	CrtReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a replication queue map	"ALTER REPLQMAP command" on page 102	AlterQReplMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a replication queue map	"DROP REPLQMAP command" on page 104	DropReplQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create peer-to-peer Q subscriptions	"CREATE QSUB command (peer-to-peer replication)" on page 131	Sample files in the folder 2nodes (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a peer-to-peer Q subscription	"ALTER QSUB command (peer-to-peer replication)" on page 132	Update0.in (located in the /sqllib/samples/repl/asnclp/ update directory)
Delete a peer-to-peer Q subscription	"DROP SUBTYPE command (peer-to-peer replication)" on page 134	delete2.in (located in the /sqllib/samples/repl/asnclp/ delete directory)
Delete a subgroup	"DROP SUBGROUP command" on page 87	DropSubGBiInput.txt (located in the /sqllib/samples/repl/asnclp/ delete directory)

ASNCLP commands for event publication

The following table lists the actions for event publishing and where you can find descriptions of the ASNCLP commands to perform those actions.

Table 6. Commands available for event publishing

Action	Command descriptions	Sample files
Create the control tables for the Q Capture program	"CREATE CONTROL TABLES command" on page 89	CrtQCaptureUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create a publishing queue map	"CREATE PUBQMAP command" on page 97	CrtPubQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter a publishing queue map	"ALTER PUBQMAP command" on page 98	AlterPubQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Delete a publishing queue map	"DROP PUBQMAP command" on page 100	DropPubQMap.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Create an XML publication	"CREATE XML PUB command" on page 105	CrtXMLPubs.txt (located in the /sqllib/samples/repl/asnclp/ directory)
Alter an XML publication	"ALTER XML PUB command" on page 108	AlterXMLUOW.txt (located in the /sqllib/samples/repl/asnclp/ directory)

Table 6. Commands available for event publishing (continued)

Action	Command descriptions	Sample files
Delete an XML publication	"DROP XML PUB command" on page 110	DropXMLPub.txt (located in the /sqllib/samples/repl/asnclp/ directory)

Replication Alert Monitor commands for SQL replication and Q replication

The Replication Alert Monitor is a utility that checks the health of programs that are part of the replication and event publishing solutions. It checks for situations in which a program terminates, issues a warning or error message, reaches a threshold for a specified value, or performs a certain action. You tell the Replication Alert Monitor which situations to watch for. If any of them occur, the Replication Alert Monitor sends an e-mail message to the person or group of persons that are designated as the appropriate contacts for such a situation. For example, you might have the Replication Alert Monitor notify you when a replication program is not running or has reached the maximum amount of memory that you expect it to use.

The following table lists the actions available for the Replication Alert Monitor program for SQL and Q replication.

Table 7. Replication Alert Monitor commands (SQL replication)

Action	Command descriptions
Create contact information for notifications	"CREATE CONTACT command" on page 143
Modify contact information for notifications	"ALTER CONTACT command" on page 144
Drop an existing contact	"DROP CONTACT command" on page 144
Substitute one existing contact with another existing contact	"SUBSTITUTE CONTACT command" on page 145
Delegate an existing contact to a new contact	"DELEGATE CONTACT command" on page 145
Create a group of replication monitor contacts	"CREATE GROUP command" on page 146
Modify a group of replication monitor contacts	"ALTER GROUP command" on page 146
Drop a group of replication monitor contacts	"DROP GROUP command" on page 147
Create alert conditions for the Q Capture program	"CREATE ALERT CONDITIONS FOR CAPTURE command" on page 149
Modify alert conditions for the Q Capture program	"ALTER ALERT CONDITIONS FOR CAPTURE command" on page 151
Drop alert conditions for the Q Capture program	"DROP ALERT CONDITIONS FOR CAPTURE command" on page 152
Create alert conditions for the Q Apply program	"CREATE ALERT CONDITIONS FOR APPLY command" on page 153
Modify alert conditions for the Q Apply program	"ALTER ALERT CONDITIONS FOR APPLY command" on page 155
Drop alert conditions for the Q Apply program	"DROP ALERT CONDITIONS FOR APPLY command" on page 157

Table 8. Replication monitor commands (Q replication)

Action	**Command description**
Create alert conditions for the Q Capture program	"CREATE ALERT CONDITIONS FOR QCAPTURE command" on page 159
Modify alert conditions for the Q Capture program	"ALTER ALERT CONDITIONS FOR QCAPTURE command" on page 161
Drop alert conditions for the Q Capture program	"DROP ALERT CONDITIONS FOR QCAPTURE command" on page 162
Create alert conditions for the Q Apply program	"CREATE ALERT CONDITIONS FOR QAPPLY command" on page 163
Modify alert conditions for the Q Apply program	"ALTER ALERT CONDITIONS FOR QAPPLY command" on page 164
Drop alert conditions for the Q Apply program	"DROP ALERT CONDITIONS FOR QAPPLY command" on page 166

Chapter 2. Using the ASNCLP program

You can run replication commands either directly from the ASNCLP command line or from an input file. The output of the commands issued using the ASNCLP program is always an SQL script, which you can run immediately or at a later time. You can issue ASNCLP commands one at a time from an operating system command prompt or you can provide multiple commands in a file that is passed as input to the ASNCLP program. For an example of running the ASNCLP program, see Chapter 6, “Subscription definition commands for SQL replication,” on page 51.

This section contains the following topics:

- “Setting up a Java environment to run the ASNCLP program”
- “Running the ASNCLP commands interactively” on page 16
- “Running the ASNCLP commands using an input file” on page 16
- “Output files from the ASNCLP program” on page 17
- “Example of creating a subscription for SQL replication” on page 17
- “Example of creating a subscription for Q replication” on page 18

Setting up a Java environment to run the ASNCLP program

Because the ASNCLP program runs in a Java™ Runtime Environment, you must add certain jar files to your Java CLASSPATH environment variable to run the ASNCLP program.

Procedure:

Add the following jar files to your Java CLASSPATH environment variable:

```
CLASSPATH = %CLASSPATH%;
    INSTDIR\sqllib\java\Common.jar;
    INSTDIR\sqllib\tools\db2cmn.jar;
    INSTDIR\sqllib\tools\db2replapis.jar;
    INSTDIR\sqllib\tools\db2qreplapis.jar;
    INSTDIR\sqllib\tools\jt400.jar;
```

where *INSTDIR* is the DB2 instance directory. On UNIX operating systems, the instance directory is the *INSTHOME*/sqllib directory, where *INSTHOME* is the home directory of the instance owner. On Windows operating systems, the instance directory is the /sqllib subdirectory where DB2 was installed.

Examples for Windows:

An example of setting the CLASSPATH environment variable from a Windows command prompt is as follows:

```
set CLASSPATH=%
CLASSPATH%;
c:\sqllib\java\Common.jar;
c:\sqllib\tools\db2cmn.jar;
c:\sqllib\tools\db2replapis.jar;
c:\sqllib\tools\db2qreplapis.jar;
c:\sqllib\tools\jt400.jar;
```

Examples for UNIX:

```
export
CLASSPATH=$CLASSPATH
:/u/myinst/sqllib/java/Common.jar;
:/u/myinst/sqllib/tools/db2cmn.jar;
:/u/myinst/sqllib/tools/db2replapis.jar;
:/u/myinst/sqllib/tools/db2qreplapis.jar;
:/u/myinst/sqllib/tools/jt400.jar;
```

Binding required when using ASNCLP with DB2 for z/OS

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

```
db2 bind @ddcsmvs.lst isolation ur blocking all
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 interactively

To run the ASNCLP program interactively (that is, manually rather than using an input file), use the ASNCLP command at the prompt to instantiate a new replication command line processor. For example:

```
ASNCLP
```

The operating-system command prompt changes to `Repl >`. From this replication command prompt, you can issue any of the ASNCLP commands.

To exit the replication command line processor, use the **quit** command. For example

```
Repl > quit
```

From the operating-system command prompt, you can get help for the ASNCLP program by using the ASNCLP command with a question mark as the only parameter. For example:

```
ASNCLP ?
```

In the command-line mode, an ASNCLP session is defined as the period from when you start the ASNCLP program and see the `Repl >` prompt for the first time until you enter the **quit** command to terminate the ASNCLP program. Anything that you do during this session applies to all the commands entered in this session.

Running the ASNCLP commands using an input file

To run the ASNCLP program in a batch mode using an input file, enter the **ASNCLP** command with -f and the input-file name as parameters at the prompt. For example:

```
ASNCLP -f myfile.in
```

The input-file name (″myfile.in″ in the example) can consist of any valid file name plus extension.

You can also specify a full file path and file name; for example:

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

The input file contains the replication administrative 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; the comments should begin with a hash (#) sign.

In the input file mode, an ASNCLP session is defined by the contents of the input file. The ASNCLP program processes all of the commands in the file until it encounters the end of the file. If the ASNCLP program encounters an error, it stops processing at that point.

Output files from the ASNCLP program

The ASNCLP program typically generates the following files:

- Log file. The output for the log file is directed to the operating system's standard output (stdout). The log file contains all the informational, warning, and error messages generated by the APIs and the ASNCLP program.
- SQL script file or files, if at least one replication command completed successfully

The names of these output files can be changed using ASNCLP environment commands.

Example of creating a subscription for SQL replication

The following example shows how to create a SQL replication subscription. To run this example, copy the code below into a file and save it as repl.in. From the Windows, UNIX, or Linux command prompt or from the db2cmd prompt, run the following command (not case sensitive):

```
 asnclp -f repl.in
```

```
# Create control tables
set output capture script "source.sql"  control script " " target script " ";
set log "cnsrc.err";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set server capture to db srcdb id regres1 password "dr0pbyme";
create control tables for capture server IN UW UOW TSUOW100 OTHERS TSASN100;

set output capture script " " control script "" target script "target.sql"  ;
set server control to db srcdb id regres1 password "dr0pbyme";
create control tables for apply control server IN UW OTHERS TSASN100;

# Create Registration
set output capture script "register.sql"  control script " " target script " ";
set log "reg.err";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set server all to db srcdb id regres1 password "dr0pbyme";
create registration (regres1.SRC001) DIFFERENTIAL REFRESH STAGE CDSRC001 cols
(C000 image both,C001 image both,C002 image after,C003 image after) prefix "X";

#Create subscription set
set output capture script " "  control script "subsset.sql" target script " " ;
set log "sstrg.err";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set server capture to db srcdb id regres1 password "dr0pbyme";
set server control to db trgdb id regres1 password "dr0pbyme";
set server target to db trgdb id regres1 password "dr0pbyme";
create subscription set setname SET00 applyqual AQ00  activate yes timing
interval 1 start date "2001-02-02" time "09:00:00.000000";

#Create Subscriptions
set output capture script "srctb.sql"
control script "control.sql"
target script "tgrtb.sql" ;
set log "smtrg.err";
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set server capture to db srcdb id regres1 password "dr0pbyme";
set server control to db trgdb id regres1 password "dr0pbyme";
set server target to db trgdb id regres1 password "dr0pbyme";

set profile tbspace for object target tablespace
options UW USING FILE "c:\TSTRG.TS"
SIZE 700 PAGES;

create member in setname SET00 applyqual AQ00
activate yes source src000 target name trg000
definition in TSTRG00 create using profile tbspace type USERCOPY
cols  all registered;
quit;
```

The **SET RUN NOW** option is used in this example. This option is necessary because subsequent actions rely on the existence of objects in previous actions. For example, if the control tables do not exist, the **CREATE REGISTRATION** command will not work.

This example produces the following output files in the same directory where you run the ASNCLP program:

- source.sql
- target.sql
- cnsrc.err
- register.sql
- reg.err
- subsset.sql
- sstrg.err
- srctb.sql
- control.sql
- tgrtb.sql
- smtrg.err

Example of creating a subscription for Q replication

The following example shows how to create a unidirectional subscription for Q replication. To run this example, copy the code below into a file and save it as repl.in. From the Windows, UNIX, or Linux command prompt or from the db2cmd prompt, run the following command (not case sensitive):

```
 asnclp -f repl.in
```

```
asnclp session set to Q replication;
# Create control tables
set output capture script "source.sql";
set log "cnsrc.err";
set qmanager "MYQMGRNAME" for capture schema;
set server capture to db srcdb ;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
create control tables for capture server using
```

```
restartq "MYRSTQNAME" adminq "MYADMINQNAME" ;

set output target script "target.sql"   ;
set qmanager "MYQMGRNAME" for apply schema;
set apply schema asn;
set server target to db trgdb ;
create control tables for apply server ;

# Create replication queue maps
set output capture script "qmapsrc.sql"
target script "qmaptrg.sql";
set log "qmap.err";
set server capture to db srcdb ;
set server target to db trgdb;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set qmanager "MYQMGRNAME" for capture schema;
create replqmap replqmap1 using adminq "MYADMINQNAME2"
recvq "Q1" sendq "Q1"
num apply agents 4 max message size 16;

#Create subscriptions
asnclp session set to Q replication;
set output capture script "dpqlob_smsrc.sql"
target script "dpqlob_smtrg.sql" ;
set log "smtrg.err";
set server capture to db srcdb ;
set server target to db trgdb ;
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
set qmanager "MYQMGRNAME" for capture schema;
set qmanager "MYQMGRNAME" for apply schema;
create qsub using replqmap replqmap1
(subname test2 src002 options has load phase I
target name TRG002  load type 2);
quit;
```

This example produces the following output files in the same directory where you run the ASNCLP program:

- source.sql
- target.sql
- cnsrc.err
- qmapsrc.sql
- qmaptrg.sql
- qmap.err
- dpqlob_smsrc.sql
- dpqlob_smtrg.sql
- smtrg.err

Part 2. Commands for SQL replication

Chapter 3. Environment commands for SQL replication

Setting up a replication environment requires you to perform a set of tasks, and these tasks might need to share the same environmental attributes. The environment commands set these attributes for the tasks. The environment commands include:

- "SET SESSION command" on page 24
- "SET SERVER command" on page 25
- "SET PROFILE command" on page 27
- "SET DROP command" on page 29
- "SET OUTPUT command" on page 31
- "SET LOG command" on page 31
- "SET CAPTURE SCHEMA command" on page 32
- "SET TRACE command" on page 32
- "SET RUN SCRIPT command" on page 33

The environment commands provide a common environment for all subsequent commands

The scope of the environment commands is limited to a single ASNCLP command-line session or to a single input file.

An environment command affects *all* of the subsequent task commands.

You can specify multiple environment commands for an ASNCLP command-line session. You can also specify the same environment command several times in a command line session; in this case, the most recent environment command overrides any previous environment commands. See Chapter 3, "Environment commands for SQL replication" for more information on the environment command.

These environment commands define the following environment attributes:

Server information: Capture control, Apply control, Monitor control, and target servers

The **SET SERVER** command removes the need for any of the task commands to refer to physical-server information and encapsulates the database connection information in one place.

For example, while creating a registration for SQL replication, setting a Capture control server to a particular database alias, will result in all subsequent task commands using that Capture server alias so that all **CREATE REGISTRATION** commands run against that database

Running the **SET SERVER** command is required before running any task command.

Note: You cannot simply use the DB2 environment variable DB2DBDFT because the ASNCLP program often needs to connect to more than one server, such as when creating subscription-set members.

Optional customization for table spaces and indexes

The **SET PROFILE** command allows you to specify table space and index information once and removes this information from the task commands.

For example, you might need to define a CD table space on z/OS with the following storage attributes: primary quantity 512 and secondary quantity 256. The **SET PROFILE** command allows you to provide that information once so that all subsequent task commands inherit these specifications.

Differences from DJRA Edit Logic: You must issue the environment commands for each session and in every input file, whereas you needed to define the DJRA Edit Logic just once. Also, the functionality provided by the environment commands is currently more limited than the DJRA Edit Logic capabilities.

Note: DJRA is mentioned here only for comparison, that is, to compare what was available for Version 7 and what is available for Version 8. Neither DJRA nor its command set are supported for Version 8.

For customizing object names, the underlying replication API uses a default naming convention so that no environment command is necessary. For example:

- Default Schema ″ASN″
- Default target table name TRG-<source table name> (for SQL replication)
- Default target table name TGT<source table name> (for Q replication)

Drop information

The **SET DROP** command allows you to set the **DROP** environment variables for tables and table spaces. These settings take effect when the task commands are invoked.

Output information

The **SET OUTPUT** and **SET LOG** commands allow you to redirect the output script and log information so that all subsequent task commands within a session inherit these specifications.

Capture schema

The **SET CAPTURE SCHEMA** command allows you to provide a Capture schema. The default Capture schema is ASN.

Trace The **SET TRACE** command allows you to enable and disable the trace.

Run option

The **SET RUN SCRIPT** command allows you to control if the replication SQL scripts should be run after they are generated.

SET SESSION command

The **SET SESSION** command specifies which type of replication this session will be used for.

Syntax

“SET SESSION” on page 200 has the dotted decimal version of the syntax diagram.

```
►►──ASNCLP SESSION SET TO──┬─SQL REPLICATION─┬──────────────────────────────────────────────►◄
                           └─Q REPLICATION───┘
```

Parameters

Specifies the type of replication this ASNCLP session will be used for.

SQL REPLICATION
Only commands for SQL replication syntax are valid and scripts to setup SQL replication are generated

Q REPLICATION
Only Q replication syntax is valid and scripts to set up Q replication are generated.

Usage notes

- Depending on the type of replication specified for this session, only syntax that applies to that type of replication will be valid.
- If this command is not issued, the default is SQL replication.

SET SERVER command

The **SET SERVER** command assigns a database alias for a logical replication server (remote iSeries source server, Capture control server, Apply control server, or target server). You can also specify a user ID and password to use when connecting to the database. The **SET SERVER** command is required for all subsequent task commands:

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

All registration commands (including promote)
Set the Capture control server before running the registration commands. For iSeries, the Remote Source server must also be set.

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

Syntax

"SET SERVER (SQL replication)" on page 199 has the dotted decimal version of the syntax diagram.

```
►►──SET SERVER──┬─ALL───────────┬─ TO──────────────────────────────────────────────────►
                ├─REMOTE SOURCE─┤
                ├─CAPTURE───────┤
                ├─CONTROL───────┤
                └─TARGET────────┘

►──┬─NULLS──────────────────────────────────────────────────────┬──────────────────►◄
   └─┬─DB dbalias──────────┬──┬─DBNAME dbname──────────┬─other-options─┘
     └─DBALIAS aliasname───┘  └─NONIBM SERVER remsrvr──┘
```

other-options:

```
├──┬───────────────────────────┬──┬─────────────┬──┬────────────────┬───────────────────────────────────────┤
   └─AS400 HOSTNAME "hostname"─┘  └─ID userid───┘  └─PASSWORD pwd───┘
```

Parameters

ALL, REMOTE SOURCE, CAPTURE, CONTROL, TARGET
Specifies which replication logical server to associate with the database alias provided:

- ALL: All servers (remote source, Capture control, Apply control, target control)
- REMOTE SOURCE: Remote source server (iSeries only)
- CAPTURE: Capture control server
- CONTROL: Apply control server
- TARGET: Target server

Subsequent task commands inherit the context that is set up by the environment command.

NULLS
Specifies that the server names should be set to NULL by the ASNCLP program. That is, this option resets any previous **SET SERVER** command.

DB, DBALIAS
Specifies the DB2 alias name (used with Linux, UNIX, or Windows).

DBNAME
Specifies the DB2 database name. For z/OS, DBNAME is the DB2 subsystem location name.

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

AS400 HOSTNAME
Specifies the OS/400 host name. This host name is typically an IP address or name.

ID Specifies the user ID to use when connecting to this server.

PASSWORD
Specifies the password to use when connecting to this server. You must specify the password in double quotation marks to preserve case sensitivity.

Usage notes

- If you include the NONIBM SERVER clause, the ASNCLP program calls the heterogeneous replication API; otherwise, the ASNCLP program calls DB2 replication APIs. "Heterogeneous" implies non-DB2 data sources such as Oracle and Sybase. The heterogeneous APIs set up replication to and from non-DB2 data sources. The environment command saves the database server information, but does not perform the actual db2 connect. The environment command assigns a database alias to a logical replication server; the ASNCLP 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, Capture control, Apply control, or target server. That is, you can associate only one value for each of these servers, but these values need not be the same.

SET PROFILE command

The **SET PROFILE** command sets up customization 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 it. You can associate a profile with a task command by specifying the profile's name in the task command.

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).

Syntax

"SET PROFILE (SQL replication)" on page 197 has the dotted decimal version of the syntax diagram.

```
►►──SET PROFILE name──┬─prof-clause─┬──────────────────────────────────────────────────────►◄
                      └─UNDO────────┘
```

prof-clause:

```
├──FOR OBJECT──┬─CD────────┬─TABLESPACE OPTIONS──┬─┤ zos-tbs-clause ├─┬──────────────────────────┤
               ├─CCD───────┤                     └─┤ uw-tbs-clause ├──┘
               ├─TARGET────┤
               ├─UOW───────┤
               ├─OTHERS────┤
               ├─PAGE LOCK─┤
               └─ROW LOCK──┘
```

zos-tbs-clause:

```
├──ZOS──┬──────────┬──┬───────────────────┬──┬──────────────────────────┬──────────────────────►
        └─DB name──┘  └─BUFFERPOOL name───┘  └─ENCODING──┬─EBCDIC──┬───┘
                                                         ├─ASCII───┤
                                                         └─UNICODE─┘

►──┬─────────────────────────────────────────────────────────────────────────────────────┬──────┤
   └─STOGROUP name──┬──────────────────────────────────┬──┬──────────────────────────────────┬─┘
                    └─PRIQTY──┬─ABSOLUTE──────────┬─ n─┘  └─SECQTY──┬─ABSOLUTE──────────┬─ m─┘
                              └─PERCENT OF SOURCE─┘                 └─PERCENT OF SOURCE─┘
```

uw-tbs-clause:

```
├──UW──┬───────────────────┬──┬─────────────┬──┬──────────────────────────────────────────────────────────────────────────┬──────────────┤
       └─BUFFERPOOL name──┘  └─PAGESIZE n─┘  └─USING──┬─FILE───┬──"container"──┬─SIZE n──┬─PAGES─┬──────┬─┘
                                                      └─DEVICE─┘                 │         ├─KILO──┤      │
                                                                                 │         ├─MEGA──┤      │
                                                                                 │         └─GIGA──┘      │
                                                                                 └─PERCENT OF SOURCE n────┘
```

Parameters

PROFILE
Specifies a profile name for this profile.

FOR OBJECT
Specifies the objects for which table space options will be set:

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
All tables that follow this locking mechanism (z/OS only)

ROW LOCK
All tables that follow this locking mechanism (z/OS only)

TABLESPACE OPTIONS
Specifies table space options. You can specify table space options for z/OS or UNIX and Windows.

Note for z/OS: No table space lock size is included because the replication API infers the correct value in most cases; the only case where the lock size might be needed is for the target table, but the environment command does not support this clause.

Notes for UNIX and Windows:

- The MANAGED BY DATABASE clause is supplied by the replication API.
- There is no support for LONG table spaces.
- There is no support for heterogeneous replication environments.

DB Specifies the database in which the tablespace will be created. This parameter does not specify the subsystem name; use the **SET SERVER** command to set the subsystem name to connect to.

BUFFER POOLS
Specifies a buffer pool name.

ENCODING
Specifies the encoding scheme. The default is EBCDIC.

STOGROUP
Specifies a storage group name.

PRIQTY
Specifies the primary quantity.

SECQTY
Specifies the secondary quantity.

ABSOLUTE
Specifies an actual value.

PERCENT OF SOURCE
Specifies percentage of the source table size, as indicated by:

- The column “npages” in SYSIBM.SYSTABLES, if the source platform is OS/390
- The column “npages” in SYSSTAT.TABLES, if the source platform is LUOW

This method will only work 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.

PAGESIZE
Specifies the page size.

USING FILE, DEVICE
Specifies the container path string. 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\. The table space name will be generated and appended to the specified path by the ASNCLP program when you run a task command such as **CREATE REGISTRATION**. The double quotation marks in the syntax are mandatory.

SIZE Specifies the number of pages for the container:

PAGES
Actual number of pages

KILO Kilobytes

MEGA
Megabytes

GIGA Gigabytes

UNDO
Nullifies the current values set for a given profile.

SET DROP command

The **SET DROP** command determines whether to drop the table space when you drop the database object (replication control tables, registrations, or subscription-set members) that it contains. Because the replication tools (ASNCLP command, Replication Center, OS/400 system commands) might not have created the table space that contains the replication object, you must decide whether you want the ASNCLP command to drop the table space when the replication object is dropped.

Note: There is an asymmetry between the drop options and the create table space options. The drop options impact multiple objects (that is, they are at the environment-command level), whereas the create options are at an object level (that is, they are at the task-command level).

Syntax

“SET DROP (SQL replication)” on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET DROP TARGET──┬─ALWAYS─┬──────────────────────────────────────────────────►◄
                     └─NEVER──┘
```

```
►►──SET DROP──┬─CD─────────────┬─TABLESPACE──┬─WHEN EMPTY─┬────────────────────►◄
              ├─CCD────────────┤             └─NEVER──────┘
              ├─TARGET─────────┤
              └─CONTROL TABLES─┘
```

Parameters

TARGET
Specifies whether the target table is to be dropped if a subscription set member is dropped or if an entire subscription set is dropped.

ALWAYS
Always drop the target table.

NEVER
Never drop the target table.

CD, CCD, TARGET, CONTROL TABLES
Specifies that the table space be dropped based on the object type:

CD Change data table

CCD Consistent-change-data table

TARGET
Target table

CONTROL TABLES
The Capture, Apply, or Monitor control tables

These options are relevant only for those operating-system environments for which the replication APIs create the table spaces. You can always specify the drop flag for each of these object types.

TABLESPACE
Specifies if the table space holding the object specified should be dropped when the object is dropped.

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

NEVER
Never drop the table space.

Usage notes

Whether to drop a target table that was autoregistered is decided implicitly by the drop subscription member API: if there are dependent subscriptions for that autoregistration, the API doesn’t drop the target table and doesn’t drop the registration; otherwise, the registration is dropped and the target table is dropped only if the **SET DROP TARGET ALWAYS** command allows it.

SET OUTPUT command

The **SET OUTPUT** command allows you to define output files for the replication command line interface. The output files contain the administration SQL statements needed to set up replication.

Syntax

“SET OUTPUT (SQL replication)” on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET OUTPUT──┬──────────────────────────┬──┬────────────────────────────┬────────────────────►
                └─CAPTURE SCRIPT "capfname"─┘  └─CONTROL SCRIPT "cntlfname"─┘

►──┬────────────────────────┬──┬──────────────────────────┬──────────────────────────────────►◄
   └─TARGET SCRIPT "trgfname"─┘  └─MONITOR SCRIPT "monfname"─┘
```

Parameters

CAPTURE SCRIPT
: Specifies the output file name for scripts to be run at the Capture server. The default file name is replcap.sql.

CONTROL SCRIPT
: Specifies the output file name for scripts to be run at the Apply control server. The default file name is replctl.sql.

TARGET SCRIPT
: Specifies the output file name for scripts to be run at the Apply server. The default file name is repltrg.sql.

MONITOR SCRIPT
: Specifies the output file name for scripts to be 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 the files already exist, the ASNCLP program will append to them.
- The double quotation marks in the command syntax are required.

SET LOG command

The **SET LOG** command allow you to define the log file for the replication command line interface. The log file contains informational messages, warnings, and errors.

Syntax

“SET LOG” on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET LOG──"logfilename"──────────────────────────────────────────────────────────────────►◄
```

Parameters

LOG *"logfilename"*
Specifies the 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.

SET CAPTURE SCHEMA command

The **SET CAPTURE SCHEMA** command allows you to set a default source and target Capture schema for all task commands. By default, the replication API uses the ASN Capture schema, so if you do not need additional or different Capture schemas, you do not need to use this command.

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

Syntax

"SET CAPTURE SCHEMA (SQL replication)" on page 195 has the dotted decimal version of the syntax diagram.

```
►►──SET CAPTURE SCHEMA──┬─SOURCE─┬──┬─TO──┬─DEFAULT─┬─┬──────────────────────────────────►◄
                        └─TARGET─┘  │     └─NULLS───┘ │
                                    └─capschema───────┘
```

Parameters

SOURCE
Specifies the Capture schema at the source. Can be any valid DB2 schema name.

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

DEFAULT
Specifies that the Capture schema will be set to ASN by the ASNCLP commands, that is, that any previous SET CAPTURE SCHEMA commands will be reset to ASN.

NULLS
Specifies that the Capture schema will be set to NULL.

SET TRACE command

The **SET TRACE** command allows you to enable and disable the trace for the ASNCLP commands. The trace is written to stdout and stderr.

Syntax

"SET TRACE" on page 200 has the dotted decimal version of the syntax diagram.

```
►►──SET TRACE──┬─OFF─┬────────────────────────────────────────────────────►◄
               └─ON──┘
```

Parameters

OFF Turns off the trace. This is the default.

ON Turns on the trace.

SET RUN SCRIPT command

The **SET RUN SCRIPT** command allows you to control whether to automatically run each task command before the ASNCLP commands process the next task command or to manually run them later in a DB2 command prompt.

This command is useful for input files that contain task commands that assume that a previous command was executed and thus that the required objects already exist in the DB2 database.

For example, if you are trying to register more than one federated source table using **CREATE REGISTRATION** commands in the input file, you should use the **SET RUN SCRIPT NOW** option.

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 the previous registration information along with the current registration. If each registration script is not executed before the next create registration, 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.

Syntax

"SET RUN SCRIPT" on page 198 has the dotted decimal version of the syntax diagram.

```
►►──SET RUN SCRIPT──┬─LATER──────────────────────────┬────────────────────────►◄
                    └─NOW──STOP ON SQL ERROR──┬─ON──┬─┘
                                              └─OFF─┘
```

Parameters

LATER

Specifies that the generated SQL scripts not be automatically executed. If you specify to run them later, the ASNCLP will not run the script. You must run generated SQL script manually at a DB2 command prompt by using this command as the following example shows:

```
db2 -tvf <filename>
```

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

NOW Specifies that the generated SQL scripts be automatically executed by the ASNCLP program.

STOP ON SQL ERROR

ON Specifies that the ASNCLP commands stop processing when the first SQL statement fails. All previous SQL statements related to this command will be rolled back. This statement is *not* valid if you are creating a subscription with different source and target servers. 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 Specifies that the ASNCLP program will not stop when it encounters SQL errors.

Usage notes

This command supports heterogeneous replication scripts.

Chapter 4. Control table definition commands for SQL replication

The **control table definition** commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command.

The **control table definition** commands include:

- "CREATE CONTROL TABLES command"
- "DROP CONTROL TABLES command" on page 38

CREATE CONTROL TABLES command

The **CREATE CONTROL TABLES** command creates a new set of Capture, Apply, or Replication Alert Monitor control tables.

This command assumes that the appropriate **SET SERVER** command was previously issued.

Syntax

"CREATE CONTROL TABLES (SQL replication)" on page 183 has the dotted decimal version of the syntax diagram.

```
►►──CREATE CONTROL TABLES FOR──┬─CAPTURE SERVER─────────┬──┬────────────────────────────────┬──────────►◄
                               ├─APPLY CONTROL SERVER───┤  └─IN───┬─ZOS── zos-ts-clause───┬─┘
                               └─MONITOR CONTROL SERVER─┘         ├─UW uw-ts-clause───────┤
                                                                  └─NONIBM fed-ts-clause──┘
```

zos-ts-clause:

```
├──┬──────────────────────────────────────────────────────────┬─────────────────────────────────────────►
   └─UOW DB dbname──┬─ tsname─────────────────┬─┤ prof-clause ├─┘
                    └─NAMING PREFIX prefix────┘

►──┬──────────────────────────────────────────────────────────┬─────────────────────────────────────────►
   └─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:

```
├──┬────────────────────────────────────────────────┬──┬──────────────────────────────────────────────────┬──────────┤
   └─UOW──┬─tsname──────────────────┬─prof-clause─┘  └─OTHERS──┬─tsname──────────────────┬─prof-clause─┘
          └─NAMING PREFIX prefix─┘                             └─NAMING PREFIX prefix─┘
```

fed-ts-clause:

```
├──┬──────────────────────────────────────────────────┬─SCHEMA name──────────────────────────────────────┤
   └─OTHERS──┬─tsname──────────────────┬─prof-clause─┘
             └─NAMING PREFIX prefix─┘
```

prof-clause:

```
├──┬───────────────────────────────┬─────────────────────────────────────────────────────────────────────┤
   ├─CREATE USING PROFILE pname─┤
   └─REUSE─────────────────────────┘
```

Parameters

CAPTURE SERVER, APPLY CONTROL SERVER, MONITOR CONTROL SERVER
Specifies the logical server to create replication control tables for.

ZOS Specifies z/OS or OS/390.

UW Specifies UNIX or Windows.

NONIBM
Specifies non-IBM heterogeneous data sources.

IN Specifies the table space. The following rules apply when using this parameter:

- The **CREATE USING PROFILE** keyword must be specified before you can use the **REUSE** keyword.
- If the **CREATE USING PROFILE** keyword is specified, then the ASNCLP uses *tsname* as the key (For z/OS, the key is *dbname.tsname*).
- If you do not specify the **IN** clause, then the command uses the DB2 defaults for table spaces.
- If the **REUSE** keyword is specified, the ASNCLP checks if the DDL object exists for the *tsname*:
  - If the DDL object exists, the flags are set as before and the fully populated DDL object is passed to the API.
  - If the DDL object does not exist, a syntax error is displayed saying that the **CREATE USING PROFILE** keyword is expected.
- If you specify the **IN** clause with a *tsname*:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** keyword and specify the name of the profile.
  - If you want to reuse an existing table space, include the **REUSE** keyword.
  - If you do not have a profile, specify the table space name with no profile, and the command assumes that the table space exists.

- If you specify the **IN** clause with a naming prefix:
  - The ASNCLP program generates the tablespace name from the prefix you specified.
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** keyword and specify the name of the profile. A table space with the generated name will be created.
  - If you want to reuse an existing table space, include the **REUSE** clause and the generated name of the table space will be reused.
  - If you do not have a profile, the generated name of the table space needs to exist.

Notes:

- You must specify the database name, even if you set the database name in the profile. This command does not create the database.
- The *tsname* input can be a heterogeneous segment or table space name.

The fully populated DDL object is passed to the API call.

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

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

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

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

SCHEMA
Specifies the remote schema name for heterogeneous replication. The default is the remote user ID. For non-IBM databases, you can specify a table space name or a segment name for those remote sources that support them.

tsname Specifies the table space for the monitor alerts table.

dbname
Specifies the name of an existing database (valid for z/OS only).

Usage notes

Because the **SET SERVER** command is required (for connectivity to the database or subsystem), you cannot create both Capture and Apply control tables in one single command.

DROP CONTROL TABLES command

The **DROP CONTROL TABLES** command drops a set of Capture, Apply, or Monitor control tables. You can use this command to drop Version 8 replication control tables and Version 7 (or earlier) replication control tables.

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

Syntax

“DROP CONTROL TABLES (SQL replication)” on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP CONTROL TABLES ON──┬─CAPTURE SERVER─────────┬──┬──────────────────────────┬──────────────►
                            ├─APPLY CONTROL SERVER───┤  └─ARCHLEVEL──┬─0801─┬──────┘
                            └─MONITOR CONTROL SERVER─┘               ├─0201─┤
                                                                     └─0805─┘

►──┬─────────────────────┬───────────────────────────────────────────────────────────────────────►◄
   └─NONIBM SCHEMA name──┘
```

Parameters

CAPTURE SERVER, APPLY CONTROL SERVER, MONITOR CONTROL SERVER
Specifies the logical server on which to drop replication 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. Contrasting with 0805 for z/OS platform, 0801 specifies control tables created on a z/OS system running in version 8 compatibility mode. This is the default.

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

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

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

- Oracle
- Sybase
- MS-SQL
- Informix®
- Teredata

Usage notes

- The **SET DROP** command affects this command.
- By default, the table spaces will not be dropped if they are empty.

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

Chapter 5. Registration definition commands for SQL replication

The registration definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context that is defined by **SET SERVER** command.

The registration definition commands include:
- "CREATE REGISTRATION command"
- "ALTER REGISTRATION command" on page 45
- "DROP REGISTRATION command" on page 48
- "PROMOTE REGISTRATION command" on page 48

CREATE REGISTRATION command

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

Syntax

"CREATE REGISTRATION" on page 186 has the dotted decimal version of the syntax diagram.

```
                                ┌─,──────────────────────────────────────────────────┐
                                ▼                                                    │
►►──CREATE REGISTRATION──(──┬────────────┬──objname──┬──────────────────────────┬───┴──)──────────►
                            └─objowner.──┘           └─RMTJRN LIB lib NAME name─┘

►──┬─DIFFERENTIAL REFRESH──┬───────────────────┬─┬──────────────────────────────────────────────►◄
   │                       └─┤ diff-ref-clause ├─┘ │
   └─FULL REFRESH ONLY───────────────────────────┘
```

diff-ref-clause:

```
├──┬────────────────────────────────────────────┬──┬──────────────────────┬──────────────────────►
   └─STAGE──┬─────────────────────┬─cd_or_ccd_name─┘  └─CONDENSED──┬─ON──┬─┘
            └─cd_or_ccd_owner──.─┘                                └─OFF─┘

►──┬───────────────────────┬──┬──────────────────────────────────────────────────────────────┬──►
   └─NONIBM──┤ fed-clause ├─┘  └─IN──┬───────────┬──┬─tsname──────────────┬─┤ prof-clause ├─┘
                                     └─DB── name─┘  └─NAMING PREFIX prefix─┘

►──┬─────────────────────────┬──┬─────────────────────────┬─────────────────────────────────────┤
   └─COLS──┤ capcol-clause ├─┘  └─OPTIONS──┤ opt-clause ├─┘
```

fed-clause:

```
├──┬──────────────────┬─remoteccdname──────────────────────────────────────────────────┤
   └─remoteccdowner.──┘
```

prof-clause:

```
├──┬─CREATE USING PROFILE pname─┬──────────────────────────────────────────────────────┤
   └─REUSE──────────────────────┘
```

capcol-clause:

```
├──ALL──IMAGE──┬─AFTER─┬──┬──────────────────────────────────────────────┬──┬─────────────────────────┬──┤
               └─BOTH──┘  │      ┌─,────────────────────────────────┐    │  └─PREFIX──┬─X──────────┬─┘
                          │      ▼                                  │    │            └─befimgpref─┘
                          └─(────colname──┬──────────────────────┬──┴─)──┘
                                          └─IMAGE──┬─AFTER─┬─────┘
                                                   └─BOTH──┘
```

opt-clause:

```
├──┬─────────────────────────────┬──┬───────────────────────────────────┬──┬─────────────────────────┬──►
   └─CONFLICT──┬─NONE─────┬──────┘  └─UPDATE AS DELETE INSERT──┬─OFF─┬──┘  └─CAPTURE──┬─ALL─────┬───┘
               ├─STANDARD─┤                                    └─ON──┘                └─CHANGES─┘
               └─ENHANCED─┘

►──┬──────────────────────┬──┬──────────────────────────┬──┬───────────────────────────┬───────────┤
   └─FORWARDING──┬─OFF─┬──┘  └─FULL REFRESH──┬─ON──┬────┘  └─STOP ON ERROR──┬─ON──┬───┘
                 └─ON──┘                     └─OFF─┘                        └─OFF─┘
```

Parameters

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

RMTJRN
LIB (the AS/400 library name); NAME (the AS/400 journal name)

DIFFERENTIAL REFRESH
Specifies that the target table will be updated periodically as the source object changes.

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

Note: If the object name is a view, then there can be multiple CD names; do not include this parameter because the replication API will generate view names for you. Thus, any values you specify for this parameter is ignored by the ASNCLP program for views.

CONDENSED

ON Specifies that the most current data value is retained. This is the default.

OFF Specifies that a history of data is retained.

Must be set to **OFF** is source non-DB2.

This parameter is ignored for a CD table, as CD tables are always noncondensed.

FULL REFRESH ONLY
Specifies that only full refresh will be done, instead of applying changes.

NONIBM
Specifies the non-IBM options.

remoteccdowner, *remoteccdname*
Specifies the CCD owner and name in the non-DB2 database.

IN Specifies the table space. The following rules apply when using this parameter:

- If you do not specify the **IN** clause, then the command uses the DB2 defaults for table spaces.
- If you specify the **IN** clause with a *tsname*:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command assumes that the table space exists.
- If you specify a naming prefix:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command creates the table space.

Notes

- For z/OS, the name includes the database name (for example, ″dbname.tsname″). You must specify the database name, even if you set the database name in the profile. This command does not create the database.
- You can specify a heterogeneous segment or table space name, but it must already exist.

capcol-clause:

COLS Specifies the columns that you want to register.

ALL Specifies that you want to register all columns.

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

IMAGE AFTER
Specifies that only after-image columns be registered.

IMAGE BOTH
Specifies that both after-image and before-image columns be registered.

PREFIX

- If you specify **IMAGE AFTER**, the prefix will be null and the target 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.

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

opt-clause:

CONFLICT
Specifies the conflict-detection level.

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

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

ENHANCED
Conflict detection that provides the best data integrity among the master and its replicas. Like 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 and keeping only the changes that originated at the master.

UPDATE AS DELETE INSERT

ON Specifies that updates are captured as delete-insert pairs.

OFF Specifies that updates are captured as updates. This is the default.

CAPTURE

ALL Specifies that everything is captured. This is the default.

CHANGES
Specifies that only changes are captured.

FORWARDING

OFF Specifies that changes from this source are not forwarded. This is the default.

ON Specifies that changes from this source are forwarded.

FULL REFRESH

ON Specifies that full refreshes are allowed for this source. This is the default.

OFF Specifies that full refreshes are not allowed for this source.

STOP ON ERROR

ON Specifies that the Capture program continues processing if it detects an error for this registration. This is the default.

OFF Specifies that the Capture program stops if it detects an error for this registration.

Usage notes

If multiple objects are registered at one time:

- The CD or CCD object owner and name clause is ignored; the replication API generates its own defaults.
- The table space specifications apply to all registrations.
- The capcol-clause defaults to ALL.
- The OPTIONS values are common across all registrations.

Example

```
create registration (PLATO.TBLONG1)
differential refresh stage PLATO.CDTBLONG;
create registration (PLATO.SUMM2)
differential refresh stage PLATO.CDSUMM2
cols ALL IMAGE BOTH prefix "X";
```

ALTER REGISTRATION command

The **ALTER REGISTRATION** command alters a registration row in the IBMSNAP_REGISTER table and allows you to add new columns to a registered source.

Syntax

"ALTER REGISTRATION" on page 179 has the dotted decimal version of the syntax diagram.

```
►►──ALTER REGISTRATION──┬─ROW──┤ row-clause ├──────┬──────────────────────────────────►◄
                        └─ADD──┤ add-cols-clause ├─┘
```

row-clause:

```
      ┌─,──────────────────────┐
      ▼                        │
├──(──┬───────────┬──objname───┴──)──┬────────────────────────┬───────────────────────►
      └─objowner.─┘                  └─CONFLICT──┬─NONE─────┬─┘
                                                 ├─STANDARD─┤
                                                 └─ENHANCED─┘
```

```
►──┬───────────────────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──────►
   └─UPDATE AS DELETE INSERT──┬─OFF─┬──┘  └─CAPTURE──┬─ALL─────┬─┘  └─FORWARDING──┬─OFF─┬─┘
                              └─ON──┘                └─CHANGES─┘                  └─ON──┘

►──┬─────────────────────────┬──┬──────────────────────────┬──────────────────────────────────────┤
   └─FULL REFRESH──┬─ON──┬───┘  └─STOP ON ERROR──┬─ON──┬───┘
                   └─OFF─┘                       └─OFF─┘
```

add-cols-clause:

```
├──┬───────────┬─objname──┬─────────────────────────────────────────────────────┬──────────────►
   └─objowner.─┘          │           ┌─,──────────────────────────────────┐    │
                          │           ▼                                     │    │
                          └─COLS──(───colname──┬─────────────────────────┬──┴─)─┘
                                               └─IMAGE──┬─AFTER──┬───────┘
                                                        ├─BOTH───┤
                                                        └─BEFORE─┘

►──┬──────────────────────────┬───────────────────────────────────────────────────────────────┤
   └─PREFIX──┬─X──────────┬───┘
             └─befimgpref─┘
```

Parameters

row-clause:

objowner, objname
Specifies the registered source to alter. You can specify multiple objects by separating them with commas.

CONFLICT
Specifies the conflict-detection level.

UPDATE AS DELETE INSERT

ON Specifies that updates are captured as delete-insert pairs.

OFF Specifies that updates are captured as updates.

CAPTURE

ALL Specifies that everything is captured.

CHANGES
Specifies that only changes are captured.

FORWARDING

ON Specifies that changes from this source are forwarded.

OFF Specifies that changes from this source are not forwarded.

FULL REFRESH

ON Specifies that full refreshes are allowed for this source.

OFF Specifies that full refreshes are not allowed for this source.

STOP ON ERROR

ON Specifies that the Capture program continues processing if it detects an error for this registration.

OFF Specifies that the Capture program stops if it detects an error for this registration.

ADD Specifies the row identified by *objowner* and *objname*. This row is the registered source to add a column to.

COLS Specifies the columns that you want to register.

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

IMAGE AFTER
Specifies that only after-image columns be registered.

IMAGE BOTH
Specifies that both after-image and before-image columns be registered.

IMAGE BEFORE
Specifies that only before-image columns be registered.

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.

After you have created a registration with **IMAGE AFTER** specified, you cannot add before-image columns to the registration. You must drop the registration and recreate it.

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

Usage notes

The parameters in this command do not have default values.

If you add a column to a CD table when the registered source also has an internal CCD table associated with it, you must:

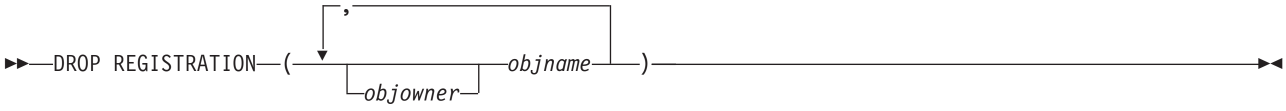
- Use the **ALTER ADD REGISTRATION COL** command to add column to CD table
- Use the **ALTER ADD SUBSCRIPTION MEMBER COL** command to add column to the internal CCD table. If you do not do this step, you will not be able to add that column to any target table that is dependent from the registered source.

DROP REGISTRATION command

The **DROP REGISTRATION** command drops one or more registrations.

Syntax

"DROP REGISTRATION" on page 193 has the dotted decimal version of the syntax diagram.



Parameters

objowner, *objname*
: Specifies the object list to drop (table, view, or nickname). You can specify multiple objects by separating them with commas.

Usage notes

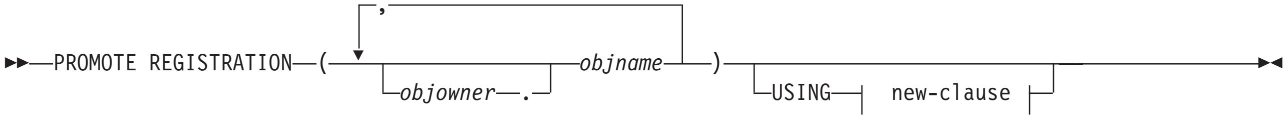
- The **SET DROP** command affects whether associated table spaces of the CD tables will be dropped when the objects are dropped.
- If the object is a view, only the CD views are dropped.
- For nicknames, this command does not drop the associated table spaces.

PROMOTE REGISTRATION command

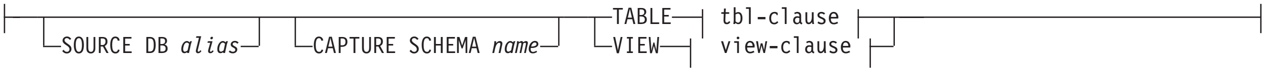
The **PROMOTE REGISTRATION** command promotes existing registrations.

Syntax

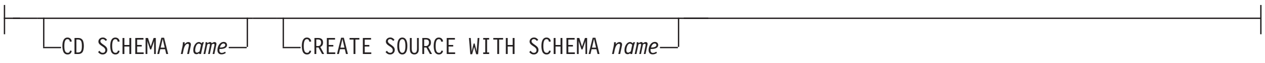
"PROMOTE REGISTRATION" on page 194 has the dotted decimal version of the syntax diagram.



new-clause:



tbl-clause:



view-clause:

```
|--+--------------------------------------------------------+--------------------------------------->
   '-CD SCHEMA FOR--+---------------+--+-----------------------+-'
                    '-VIEW name-----'  '-SOURCE TABLE name-----'

>--+------------------------------------------------------------------------------+----------------|
   '-CREATE SOURCE VIEW--+---------------------------------+--USING SCHEMA name-'
                         '-WITH UNREGISTERED BASE TABLES---'
```

Parameters

objowner, *objname*
: Specifies the objects to promote (tables or views). You can specify multiple objects by separating them with commas.

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

CAPTURE SCHEMA
: If the Capture schema is not provided when promoting a registration, the Capture schema of the registration being promoted is used.

CD SCHEMA FOR TABLE
: Specifies the new CD-table schema name for the promoted object.

CREATE SOURCE WITH SCHEMA
: Specifies the new source-table schema name to use when promoting the underlying table.

CD SCHEMA FOR VIEW
: Specifies the new CD-view schema name for the promoted object.

SOURCE TABLE
: Specifies the new CD-table schema name for the promoted object.

CREATE SOURCE VIEW
: Specifies that you want to promote the view on the new source.

WITH UNREGISTERED BASE TABLES
: Specifies that you want to promote underlying base tables that are not registered.

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

Usage notes

- If you do not specify the **USING** new-clause 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 ASNCLP command does not support a new source CD schema when promoting subscription sets, so do not change the CD schema when promoting registrations.

Chapter 6. Subscription definition commands for SQL replication

The subscription set definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command.

The subscription definition commands include:
- “CREATE SUBSCRIPTION SET command”
- “ALTER SUBSCRIPTION SET command” on page 53
- “CREATE MEMBER command” on page 54
- “DROP MEMBER command” on page 61
- “ALTER MEMBER ADD COLS command” on page 62
- “CREATE STMT command” on page 63
- “DROP STMT command” on page 64
- “DROP SUBSCRIPTION SET command” on page 65
- “PROMOTE SUBSCRIPTION SET command” on page 65

CREATE SUBSCRIPTION SET command

The **CREATE SUBSCRIPTION SET** command creates an empty subscription set.

Syntax

“CREATE SUBSCRIPTION SET” on page 191 has the dotted decimal version of the syntax diagram.

```
►►──CREATE SUBSCRIPTION SET──SETNAME── setname──APPLYQUAL── applyq──┬──────────────────────┬──────────►
                                                                    └─ACTIVATE──┬─NO───┬─┘
                                                                                ├─YES──┤
                                                                                └─ONCE─┘

►──┬─────────────┬──┬──────────────────────────────────────────────┬──────────────────────────────────►
   └─SETTYPE──┬─R─┘  └─TIMING──┬─EVENT── name──────────────────────┬─┘
              ├─U─┤            ├─INTERVAL──mn──────────────────────┤
              └─P─┘            ├─BOTH──EVENT name─ INTERVAL mn─────┤
                               └─CONTINUOUS────────────────────────┘

►──┬─────────────────────────────────────────────────────┬──┬─────────────────────────────────┬────────►
   └─START DATE "yyyy-mm-dd" TIME "hh:mm:ss.ffffff" ─────┘  └─NONIBM SOURCE SERVER srvrname──┘

►──┬────────────────────┬──┬──────────────────┬──────────────────────────────────────────────────────►◄
   └─BLOCKING minutes───┘  └─COMMIT COUNT n───┘
```

Parameters

SETNAME
Specifies the subscription-set name.

APPLYQUAL
Specifies the Apply qualifier for the subscription set.

ACTIVATE
Specifies whether to activate the subscription set.

NO Specifies that the subscription set be deactivated. This is the default.

YES Specifies that the subscription set be activated.

ONCE Specifies that the subscription set be activated for one Apply cycle, then deactivated.

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
Specifies the event, which when posted to the IBMSNAP_SUBS_EVENT table, causes the Apply program to process the subscription set.

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

BOTH Specifies that this subscription set use 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
Specifies the date when the subscription should be active. The double quotation marks are required.

TIME Specifies the time when the subscription should be active. The double quotation marks are required.

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

BLOCKING
Specifies a threshold limit to regulate the amount of data to fetch and apply. The default value is 30 minutes. This keyword controls the MAX_SYNCH_MINUTES column of the IBMSNAP_SUB_SET table.

COMMIT COUNT
Specify 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.

ALTER SUBSCRIPTION SET command

The **ALTER SUBSCRIPTION SET** command alters certain values for a subscription set.

Syntax

"ALTER SUBSCRIPTION SET" on page 180 has the dotted decimal version of the syntax diagram.

```
►►──ALTER SUBSCRIPTION SET──SETNAME── setname──APPLYQUAL── applyq──SETTYPE──┬─R──────────────────┬──────────►
                                                                            ├─U──────────────────┤
                                                                            │     ┌─F ONLY─┐     │
                                                                            │     └─S ONLY─┘     │
                                                                            └─P──────────────────┘

►──┬────────────────────┬──┬─────────────────────────────────────────────┬──┬───────────────────┬─────────►
   └─ACTIVATE──┬─NO───┬─┘  └─TIMING──┬─EVENT── name──────────────────────┬─┘  └─BLOCKING minutes─┘
               ├─YES──┤              ├─INTERVAL──mn──────────────────────┤
               └─ONCE─┘              ├─BOTH──EVENT name─ INTERVAL mn─────┤
                                     └─CONTINUOUS────────────────────────┘

►──┬──────────────────────────┬──────────────────────────────────────────────────────────────────────────►◄
   └─COMMIT COUNT──┬─ n───┬───┘
                   └─NULL─┘
```

Parameters

SETNAME
Specifies the subscription-set name.

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 If no option is specified, the default is an update-anywhere set in both F and S directions.

F ONLY
Specifies an update-anywhere set in the F direction only.

S ONLY
Specifies an update-anywhere set in the S direction only.

P Specifies a peer-to-peer set.

ACTIVATE
Specifies whether to activate the subscription set.

NO Specifies that the subscription set be deactivated.

YES Specifies that the subscription set be activated.

ONCE Specifies that the subscription set be activated for one Apply cycle, then deactivated.

TIMING
Specifies the timing for the subscription set.

EVENT
Specifies the event, which when posted to the IBMSNAP_SUBS_EVENT table, causes the Apply program to process the subscription set.

INTERVAL
Specifies the interval for the Apply program to process the subscription set.

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

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

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

COMMIT COUNT
Specify 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 issues just one COMMIT statement for the subscription set after it processes the entire set.

CREATE MEMBER command

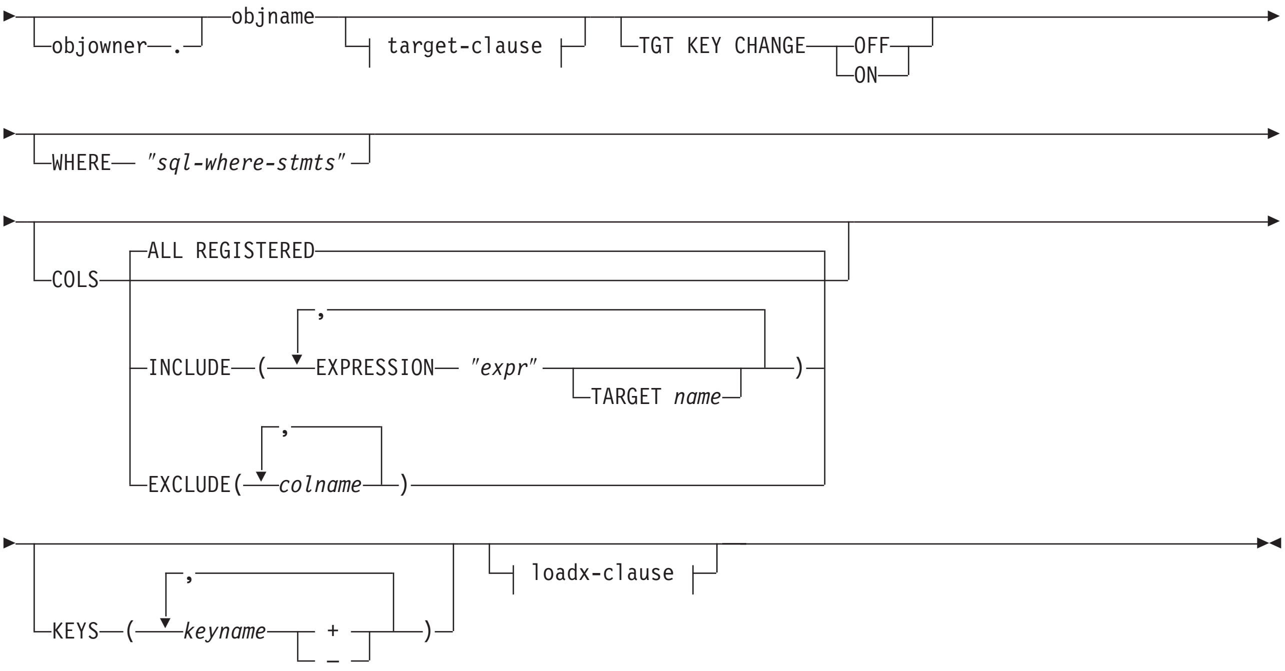
The **CREATE MEMBER** command adds a subscription-set member to an existing subscription set. Adding a member to a set includes:

- 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 IS_KEY value for the index.

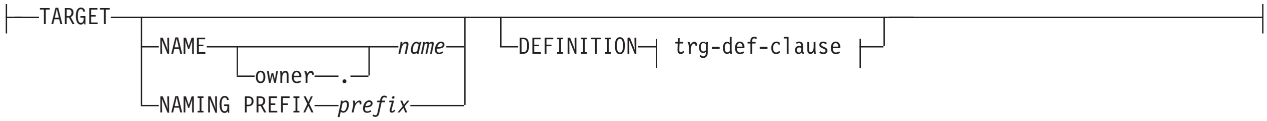
Syntax

"CREATE MEMBER" on page 184 has the dotted decimal version of the syntax diagram.

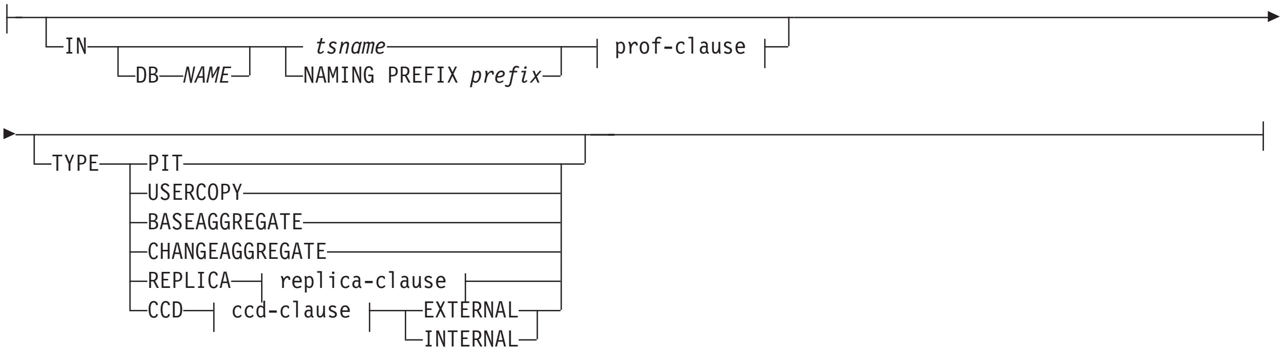
```
►►──CREATE MEMBER──IN──SETNAME── setname──APPLYQUAL── applyq──┬──────────────────────┬──SOURCE──────────►
                                                              └─ACTIVATE──┬─NO───┬─┘
                                                                          ├─YES──┤
                                                                          └─ONCE─┘
```



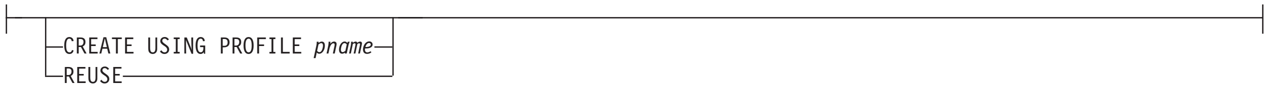
target-clause:



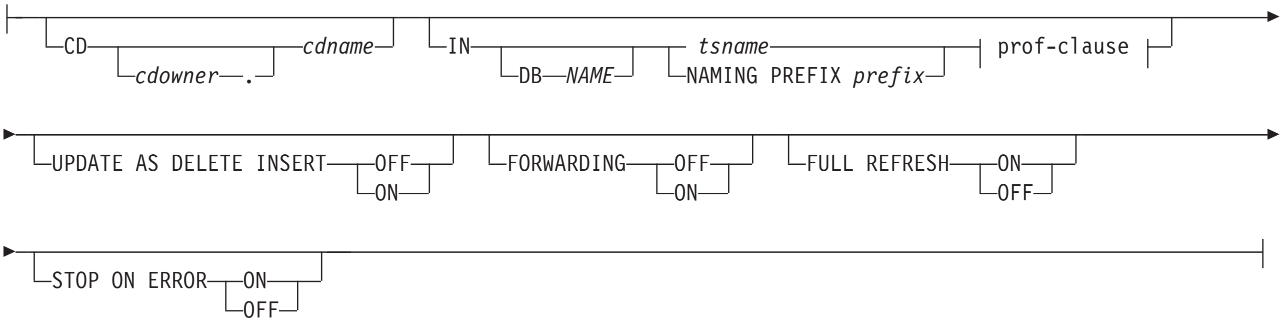
trg-def-clause:



prof-clause:

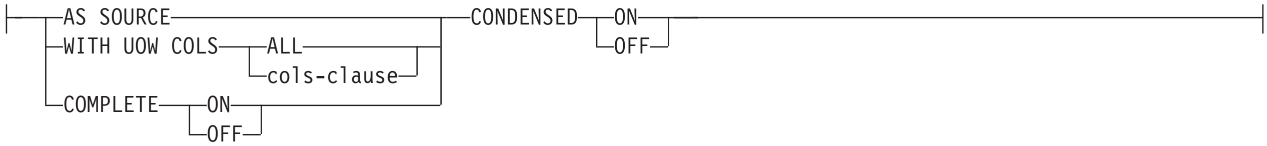


replica-clause:



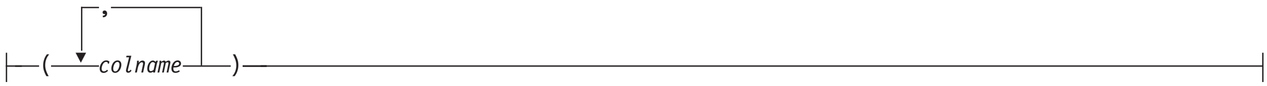
```
|──┬──────────────────────────────┬──┬──────────────────────────────────────────────────┬──┬──────────────────────┬──►
   └─CD──┬──────────────┬─cdname──┘  └─IN──┬────────────┬──┬─tsname─────────────────┬─┘  └─| prof-clause |─┘
         └─cdowner──.───┘                  └─DB──NAME───┘  └─NAMING PREFIX prefix───┘

►──┬──────────────────────────────────┬──┬───────────────────────┬──┬─────────────────────────┬──►
   └─UPDATE AS DELETE INSERT──┬─OFF─┬─┘  └─FORWARDING──┬─OFF─┬───┘  └─FULL REFRESH──┬─ON──┬───┘
                              └─ON──┘                  └─ON──┘                      └─OFF─┘

►──┬───────────────────────────┬───────────────────────────────────────────────────────────────|
   └─STOP ON ERROR──┬─ON──┬────┘
                    └─OFF─┘
```

ccd-clause:



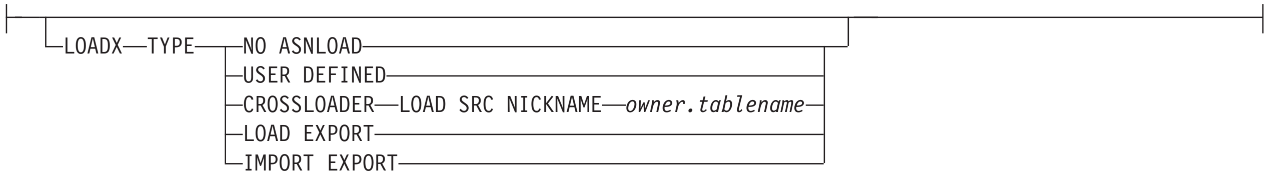
```
|──┬─AS SOURCE─────────────────────────────┬──CONDENSED──┬─ON──┬──────────────────────────────|
   ├─WITH UOW COLS──┬─ALL───────────┬──────┤             └─OFF─┘
   │                └─cols-clause───┘      │
   └─COMPLETE──┬─ON──┬─────────────────────┘
               └─OFF─┘
```

cols-clause:



```
        ┌─,──────┐
        ▼        │
|──(────colname──┴──)──────────────────────────────────────────────────────────────────────────|
```

loadx-clause:



```
|──┬──────────────────────────────────────────────────────────────────────────┬────────────────|
   └─LOADX──TYPE──┬─NO ASNLOAD────────────────────────────────────────────┬───┘
                  ├─USER DEFINED──────────────────────────────────────────┤
                  ├─CROSSLOADER──LOAD SRC NICKNAME──owner.tablename───────┤
                  ├─LOAD EXPORT───────────────────────────────────────────┤
                  └─IMPORT EXPORT─────────────────────────────────────────┘
```

Parameters

SETNAME
Specifies the subscription-set name.

APPLYQUAL
Specifies the Apply qualifier for the subscription set.

ACTIVATE
Specifies whether to activate the subscription set.

NO Specifies that the subscription set be deactivated. This is the default.

YES Specifies that the subscription set be activated.

ONCE Specifies that the subscription set be activated for one Apply cycle, then deactivated.

SOURCE
Specifies the source object. The source owner is optional.

TARGET
Specifies the target object.

NAME
Specifies the target object. The target owner and name are optional.

NAMING PREFIX
Specifies a prefix to be used to generate a target-table name.

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

IN Specifies the table space. The following rules apply when using this parameter:

- If you do not specify the **IN** clause, then the command uses the DB2 defaults for table spaces.
- If you specify the **IN** clause with a *tsname* value:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command assumes that the table space exists.
- If you specify the **IN** clause with a naming prefix:
  - ASNCLP generates the table space name using the naming prefix.
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command creates the table space with the generated name.
- The **CREATE USING PROFILE** clause must be specified before you can use the REUSE clause.
- If the **CREATE USING PROFILE** clause is specified, then the ASNCLP uses the value of *tsname* as the key (for z/OS, the key is *dbname.tsname*).
- If the **REUSE** clause is specified, the ASNCLP checks if the DDL object exists for the table space specified in *tsname*:
  - If the DDL object exists, the flags are set as before and the fully populated DDL object is passed to the API.
  - If the DDL object does not exist, a syntax error is displayed saying that the **CREATE USING PROFILE** clause is expected.

Notes

- For z/OS, the name includes the database name (for example, ″dbname.tsname″). You must specify the database name, even if you set the database name in the profile. This command does not create the database.
- You can specify a heterogeneous segment or table space name, but it must already exist.

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 (that is, the CD or an internal CCD table).

REPLICA
Specifies a replica table for update-anywhere replication.

CD Specifies *cdowner* and *cdname* (the object names for the CD table for the replica table).

IN Specifies the table space. The following rules apply when using this parameter:

- If you do not specify the **IN** clause, then the command uses the DB2 defaults for table spaces.
- If you specify the **IN** clause with a *tsname* value:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command assumes that the table space exists.
- If you specify the **IN** clause with a naming prefix:
  - ASNCLP generates the table space name using the naming prefix.
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you do not have a profile, specify the table space name with no profile, and the command creates the table space with the generated name.
- The **CREATE USING PROFILE** clause must be specified before you can use the REUSE clause.
- If the **CREATE USING PROFILE** clause is specified, then the ASNCLP uses the value of *tsname* as the key (for z/OS, the key is *dbname.tsname*).
- If the **REUSE** clause is specified, the ASNCLP checks if the DDL object exists for the *tsname* value:
  - If the DDL object exists, the flags are set as before and the fully populated DDL object is passed to the API.

- If the DDL object does not exist, a syntax error is displayed saying that the **CREATE USING PROFILE** clause is expected.

Notes

- For z/OS, the name includes the database name (for example, ″dbname.tsname″). You must specify the database name, even if you set the database name in the profile. This command does not create the database.
- You can specify a heterogeneous segment or table space name, but it must already exist.

UPDATE AS DELETE INSERT
Specifies how to handle SQL UPDATE statements.

ON Specifies that updates are captured as delete-insert pairs.

OFF Specifies that updates are captured as updates. This is the default.

FORWARDING
Specifies whether to forward captured changes to other replicas.

ON Specifies that captured changes are forwarded.

OFF Specifies that captured changes are not forwarded. This is the default.

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

ON Specifies that a full refresh be performed. This is the default.

OFF Specifies that a full refresh be performed.

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

ON Specifies that the Capture program is to stop if a Capture error occurs. This is the default.

OFF Specifies that the Capture program is to continue if a Capture error occurs.

CCD Specifies a consistent change data (CCD) table.

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.

cols-clause
Specifies which of the UOW columns should be included in the CCD table.

colname
Specifies the specific columns from the UOW table that the CCD table includes These columns are: IBMSNAP_APPLY_QUAL, IBMSNAP_AUTHID, IBMSNAP_AUTHTKN, IBMSNAP_REJ_CODE, and IBMSNAP_UOWID.

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 the CCD table is condensed.

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.

EXTERNAL
Specifies that the CCD table is external.

INTERNAL
Specifies that the CCD table is internal.

loadx-clause:

LOADX TYPE
Specifies the type of load to be used with this member.

NO ASNLOAD
Specifies that ASNLOAD will *not* be used for this member.

USER DEFINED
Specifies that a user-defined or user-modified ASNLOAD exit code will be used.

CROSSLOADER
Specifies that the crossloader utility will be used for this member.

LOAD SRC NICKNAME
Both *owner* and *tablename* are required.

LOAD EXPORT
Specifies that an EXPORT/LOAD combination will be used for this member. This keyword is used for UNIX and Windows only.

IMPORT EXPORT
Specifies that an EXPORT/IMPORT combination will be used for this member. This keyword is used for UNIX and Windows only.

TGT KEY CHANGE
Specifies whether the target key can change.

ON The key value can change.

OFF The key value cannot change. This is the default.

WHERE
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
Include all registered columns.

INCLUDE
Specifies the columns to include.

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

TARGET
Specifies the name of the target column.

EXCLUDE
Exclude the specified columns.

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

Usage notes

- The target object is not required for the command line, but the API does require a target object so that the command line 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.

DROP MEMBER command

The **DROP MEMBER** command drops a member from an existing subscription set.

Syntax

"DROP MEMBER" on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP MEMBER──FROM──SETNAME── setname──APPLYQUAL── applyq──SOURCE──┬──────────────┬──objname──────────►
                                                                      └─objowner──.──┘

►─TARGET──┬──────────────┬──objname────────────────────────────────────────────────────────────────────►◄
          └─objowner──.──┘
```

Parameters

SETNAME
Specifies the subscription-set name.

APPLYQUAL
Specifies the Apply qualifier for the subscription set.

SOURCE
Specifies the source object.

TARGET
Specifies the target object.

Usage notes

- For update-anywhere subscription sets, members for both replication directions (master-to-replica and replica-to-master) are dropped.
- 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. However, if the target table was created as part of autoregistration (a replica table or CCD table):
  - 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 command. The autoregistration information is deleted.

ALTER MEMBER ADD COLS command

The **ALTER MEMBER ADD COLS** command adds columns to an existing member in an existing subscription set.

Syntax

"ALTER MEMBER ADD COLS" on page 176 has a dotted decimal version of the syntax diagram.

```
►►──ALTER MEMBER ADD COLS──IN──SETNAME── setname──APPLYQUAL── applyq──SOURCE──────────────────►

►─┬──────────────┬─objname──TARGET─┬──────────────┬─objname──────────────────────────────────►
  └─objowner──.─┘                  └─objowner──.─┘

►─COLS──(──EXPRESSION "source-col-or-expr"─┬────────────────────┬─)─────────────────────────►◄
                                           └─TARGET name─┬───┬─┘
                                                         └─+─┘
```

Parameters

SETNAME
Specifies the subscription-set name.

APPLYQUAL
Specifies the Apply qualifier for the subscription set.

SOURCE
Specifies the source object.

TARGET
Specifies the target object.

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

EXPRESSION
Specifies an expression for the column. The double quotation marks are required.

TARGET *name*
Specifies the column name for the target.

+ 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.

CREATE STMT command

The **CREATE STMT** command creates a statement for an existing subscription set.

Syntax

"CREATE STMT" on page 190 has the dotted decimal version of the syntax diagram.

```
►►──CREATE STMT──IN──SETNAME── setname──APPLYQUAL── applyq──┬──────────────────┬────────────────►
                                                            └─SETTYPE──┬─R─┬──┘
                                                                       ├─U─┤
                                                                       └─P─┘

  ┌─SQL "statement"──┐
►─┤                  ├─┬─────────────────────────┬─EXECUTE──┬─AT SOURCE────────┬─────────────►
  └─PROC── "procname"┘ └─NUMBER── stmtnumber─────┘          ├─AFTER AT TARGET──┤
                                                            └─BEFORE AT TARGET─┘

►─┬──────────────────────┬──────────────────────────────────────────────────────────────────►◄
  └─SQLSTATES "states"───┘
```

Parameters

SETNAME
Specifies the subscription-set name.

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

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

NUMBER
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
Execute the statement or procedure at the source server.

AFTER AT TARGET
Execute the statement or procedure at the target server after the Apply program processes the subscription set.

BEFORE AT TARGET
Execute the statement or procedure at the target server before the Apply program processes the subscription set.

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

DROP STMT command

The **DROP STMT** command drops statements from an existing subscription set.

Syntax

"DROP STMT" on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP STMT──FROM──SETNAME── setname──APPLYQUAL── applyq──┬──────────────────┬──────────────────►
                                                            └─SETTYPE──┬─R─┬──┘
                                                                       ├─U─┤
                                                                       └─P─┘


              ┌─,─────────┐
              ▼           │
►─NUMBER──(──── number────┴──)─────────────────────────────────────────────────────────────────►◄
```

Parameters

SETNAME
Specifies the subscription-set name.

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

Usage notes

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

DROP SUBSCRIPTION SET command

The **DROP SUBSCRIPTION SET** command drops an existing subscription set for a specified Apply qualifier.

Syntax

"DROP SUBSCRIPTION SET" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──DROP SUBSCRIPTION SET──SETNAME── setname──APPLYQUAL── applyq──────────────────────────────►◄
```

Parameters

SETNAME
Specifies the subscription-set name.

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 61 command for the rules that affect the dropped objects.

PROMOTE SUBSCRIPTION SET command

The **PROMOTE SUBSCRIPTION SET** command promotes an existing subscription set.

Syntax

"PROMOTE SUBSCRIPTION SET" on page 195 has the dotted decimal version of the syntax diagram.

```
►►──PROMOTE SUBSCRIPTION SET──SETNAME setname──APPLYQUAL── applyq──┬──────────────────┬──────►◄
                                                                   └─USING new-clause─┘
```

new-clause:

```
►►──┬──────────────────────────────────────────────────────────────┬──────────────────────────►
    └─CAPTURE SCHEMA FOR──┬───────────────┬──┬─────────────────┬───┘
                          └─SOURCE name───┘  └── REPLICA name──┘
```

```
►─┬──────────────────────────────────────────────────────────────────────┬──┬─────────────────────┬──────────────►
  └─DB FOR──┬──────────────┬──┬────────────────┬──┬─────────────────┬─┘  └─APPLYQUAL name──┘
            └─SOURCE alias─┘  └─ TARGET── alias─┘  └─ CONTROL── alias─┘

►─┬───────────────┬──┬────────────────────┬──┬──────────────────────────────────────────────┬────────────────►◄
  └─SETNAME name──┘  └─SOURCE SCHEMA name─┘  └─ TARGET──┬───────────────┬──┬──────────────────┬─┘
                                                        └─SCHEMA name───┘  └─CD SCHEMA name───┘
```

Parameters

SETNAME
Specifies the subscription-set name.

APPLYQUAL
Specifies the Apply qualifier for the subscription set.

USING
Specifies the information for the promoted subscription set.

CAPTURE SCHEMA FOR
Specifies the new Capture schema.

SOURCE
Specifies the new Capture schema at the source.

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

DB FOR
Specifies the new database alias.

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

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

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

APPLYQUAL
Specifies the new Apply qualifier.

SETNAME
Specifies the new subscription-set name.

SOURCE SCHEMA
Specifies the new source schema name.

TARGET
Specifies the schemas for the target.

SCHEMA
Specifies the new target schema name.

CD SCHEMA
Specifies the new target-CD schema name.

Usage notes

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

Chapter 7. Offline load utility commands for SQL replication

The **OFFLINE LOAD** command is a task command that executes within the context of the replication command-line interface. It inherits the context defined by **SET SERVER** command.

OFFLINE LOAD command

The **OFFLINE LOAD** command allows you to control a manual full refresh for offline load procedures. The **OFFLINE LOAD** command helps you change the control the replication environment by changing the control tables. 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 SET and IBMSNAP_SIGNAL tables

Syntax

“OFFLINE LOAD” on page 194 has the dotted decimal version of the syntax diagram.

```
►►──OFFLINE LOAD──┬─BEFORE─┬──SETNAME setname──APPLYQUAL applyq──────────────────────────────►◄
                  └─AFTER──┘
```

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 *applyq*
Specifies the Apply qualifier for the subscription set.

Part 3. Commands for Q replication and event publishing

Chapter 8. Environment commands for unidirectional Q replication

Setting up a replication environment requires you to perform a set of tasks, and these tasks might need to share the same environmental attributes. The Q replication environment commands set these attributes for the tasks. The Q replication environment commands include:

- “SET SESSION command”
- “SET SERVER command” on page 74
- “SET QMANAGER command” on page 74
- “SET PROFILE command” on page 75
- “SET DROP command” on page 78
- “SET OUTPUT command” on page 78
- “SET LOG command” on page 80
- “SET TRACE command” on page 80
- “SET RUN SCRIPT command” on page 80
- “SET CAPTURE SCHEMA command” on page 81
- “SET APPLY SCHEMA command” on page 82
- “SHOW SET ENV command” on page 82

SET SESSION command

The **SET SESSION** command establishes a session for your chosen type of replication.

Syntax

“SET SESSION” on page 200 has the dotted decimal version of the syntax diagram.

```
►►──ASNCLP SESSION SET TO──┬─SQL REPLICATION─┬──────────────────────────────────────────────────►◄
                           └─Q REPLICATION───┘
```

Parameters

Specifies the type of replication this ASNCLP session will be used for.

SQL REPLICATION
Only commands for SQL replication syntax are valid and scripts to setup SQL replication are generated.

Q REPLICATION
Only Q replication syntax is valid and scripts to set up Q replication are generated.

Usage notes

- This must be the first command you issue in this session; otherwise, the ASNCLP program will assume SQL replication.
- Depending on the type of replication specified for this session, only syntax that applies to that type of replication will be valid.

SET SERVER command

The **SET SERVER** command allows you to specify the server (database) used in the ASNCLP session. You can specify authentication information and other required parameters for connecting to the server.

Syntax

"SET SERVER (unidirectional Q replication)" on page 200 has the dotted decimal version of the syntax diagram.

```
►►──SET SERVER──┬─CAPTURE─┬──TO──┬─NULLS──────────────────────────────────────────────────────────────┬────────────►◄
                └─TARGET──┘      └─┬─DB dbalias────────┬──┬─────────────────────┬──other-options──┘
                                   └─DBALIAS aliasname─┘  └─DBNAME dbname───────┘
```

other-options:

```
├──┬───────────┬──┬──────────────┬──────────────────────────────────────────────────────────────────┤
   └─ID userid─┘  └─PASSWORD pwd─┘
```

Parameters

CAPTURE, TARGET
Specifies the type of server being associated with the database alias:

- CAPTURE: Q Capture control server
- TARGET: Target server

NULLS
Specifies that the server names are to be set to NULL. This option resets previously set servers.

DB, DBALIAS
Specifies the DB2 alias name (used with Linux, UNIX, or Windows).

DBNAME
Specifies the DB2 database name (used with z/OS).

ID Specifies the user ID to use when connecting to this server.

PASSWORD
Specifies the password to use when connecting to this server.

SET QMANAGER command

The **SET QMANAGER** command sets the WebSphere queue manager.

Syntax

"SET QMANAGER" on page 198 has the dotted decimal version of the syntax diagram.

```
►►──SET QMANAGER── "mgrname" ──FOR──┬─CAPTURE SCHEMA──────────────────────┬──────────────────────────►◄
                                    ├─APPLY SCHEMA────────────────────────┤
                                    └─MULTIDIR servername.schemaname──────┘
```

Parameters

"mgrname"
: Name of the WebSphere queue manager to be set.

FOR

CAPTURE SCHEMA
Set this queue manager for the Q Capture control tables.

APPLY SCHEMA
Set this queue manager for the Q Apply control tables.

MULTIDIR
Set this queue manager for the bidirectional or peer-to-peer replication node.

- *servername*: Name of the server (database) using this queue manager.
- *schemaname*: Name of the replication catalog tables (see "SET MULTIDIR SCHEMA command" on page 84) using this queue manager.

This set method is optional.

SET PROFILE command

The **SET PROFILE** command allows you to specify custom parameters for database objects to be created implicitly. 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 (Q replication)" on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET PROFILE name──┬─prof-clause─┬────────────────────────────────────────────►◄
                      └─UNDO────────┘
```

prof-clause:

```
├──FOR OBJECT──┬─TARGET─────┬──┬─TABLESPACE OPTIONS──┤ zos-tbs-clause ├──┬─┬──────────────┤
               ├─QCNTL TBLS─┤  │                     └┤ uw-tbs-clause ├──┘ │
               ├─PAGE LOCK──┤  └─INDEX OPTIONS────┤ zos-idx-clause ├───────┘
               └─ROW LOCK───┘
```

zos-tbs-clause:

```
├──ZOS──┬─────────┬──┬──────────────────┬──┬──────────────────────┬──────────────────►
        └─DB name─┘  └─BUFFERPOOL name──┘  └─ENCODING──┬─EBCDIC──┬─┘
                                                       ├─ASCII───┤
                                                       └─UNICODE─┘
```

```
►─┬──────────────────────────────────────────────────────────────────────────────────────────┬──────────┤
  └─STOGROUP name─┬──────────────────────────────────────┬─┬──────────────────────────────────────┬─┘
                  └─PRIQTY──┬─ABSOLUTE n─────────────────┬─┘ └─SECQTY──┬─ABSOLUTE m─────────────────┬─┘
                            ├─PERCENT OF SOURCE n────────┤              ├─PERCENT OF SOURCE m────────┤
                            └─PERCENT OF SOURCE ALLOC n──┘              └─PERCENT OF SOURCE ALLOC m──┘
```

uw-tbs-clause:

```
├──UW──┬────────────────────┬──┬──────────────┬──────────────────────────────────────────────────────►
       └─BUFFERPOOL name────┘  └─PAGESIZE n───┘

►─┬──────────────────────────────────────────────────────────────────────┬──────────────────────────┤
  └─USING──┬─FILE───┬──"container"───┬─SIZE n──┬─PAGES─┬────────────────┬─┘
           └─DEVICE─┘                │         ├─KILO──┤                │
                                     │         ├─MEGA──┤                │
                                     │         └─GIGA──┘                │
                                     ├─PERCENT OF SOURCE n──────────────┤
                                     └─PERCENT OF SOURCE ALLOC m────────┘
```

zos-idx-clause:

```
├──ZOS──┬──────────────────┬────────────────────────────────────────────────────────────────────────►
        └─BUFFERPOOL name──┘

►─┬──────────────────────────────────────────────────────────────────────────────────────────┬──────────┤
  └─STOGROUP name─┬──────────────────────────────────────┬─┬──────────────────────────────────────┬─┘
                  └─PRIQTY──┬─ABSOLUTE n─────────────────┬─┘ └─SECQTY──┬─ABSOLUTE m─────────────────┬─┘
                            ├─PERCENT OF SOURCE n────────┤              ├─PERCENT OF SOURCE m────────┤
                            └─PERCENT OF SOURCE ALLOC n──┘              └─PERCENT OF SOURCE ALLOC m──┘
```

Parameters

PROFILE
Specifies a profile name for this profile.

FOR OBJECT
Specifies the objects for which table space options will be set:

TARGET
Target table

QCNTL TBLS
Q replication control tables

PAGE LOCK
All tables that follow this locking mechanism (z/OS only)

ROW LOCK
All tables that follow this locking mechanism (z/OS only)

TABLESPACE OPTIONS
Specifies table space options. You can specify table space options for z/OS, Linux, UNIX, and Windows.

INDEX OPTIONS
Tells the ASNCLP program that you are specifying index options rather than tablespace options.

DB Specifies the z/OS logical database name to connect to.

BUFFERPOOL
Specifies a buffer pool name. This keyword is optional.

ENCODING
Specifies the encoding scheme. The default is EBCDIC.

STOGROUP
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
The number (denoted as *n* or *m* in the syntax diagram) specifies the space allocation is at least that number of kilobytes. If it is used in conjunction with the **PRIQTY** keyword, it specifies the minimum primary space allocation. If used in conjunction with the **SECQTY** keyword, it specifies the minimum secondary space allocation. See the information **CREATE TABLESPACE** command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PERCENT OF SOURCE
The number (denoted as *n* or *m* in the syntax diagram) specifies the space allocation is at least that percentage of the current table space usage of the related source table. If it is used in conjunction with the **PRIQTY** keyword, it specifies the minimum primary space allocation. If used in conjunction with the **SECQTY** keyword, it specifies the minimum secondary space allocation. See the information **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 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, it specifies the minimum primary space allocation. If used in conjunction with the **SECQTY** keyword, it specifies the minimum secondary space allocation. See the information **CREATE TABLESPACE** command in the *DB2 UDB for z/OS V8 SQL Reference* (SC18-7426-00) for more details.

PAGESIZE
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.

USING FILE, DEVICE
Specifies the container path string. 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\.

SIZE 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.

SET DROP command

The **SET DROP** command determines whether to drop the table space when you drop the replication object. This command does not apply to tables and table spaces created in bidirectional or peer-to-peer replication.

Syntax

“SET DROP (Q replication)” on page 196 has the dotted decimal version of the syntax diagram.

```
►►—SET DROP TARGET——┬NEVER——┬——————————————————————————————————————————————————►◄
                    └ALWAYS—┘
```

```
►►—SET DROP——┬TARGET—————————┬TABLESPACE——┬WHEN EMPTY—┬——————————————————————►◄
             └CONTROL TABLES—┘             └NEVER——————┘
```

Parameters

TARGET
Specifies if the target table is to be dropped with the subscription. The default is **NEVER**.

TARGET, CONTROL TABLES

TARGET
Specifies that this setting is for the target table’s table space.

CONTROL TABLES
Specifies that this setting is for the control table’s table space.

TABLESPACE

WHEN EMPTY
Only drop table spaces if no tables depend on it.

NEVER
Do not drop table spaces.

Usage notes

Tables and table spaces created or used in bidirectional or peer-to-peer replication are never dropped implicitly.

SET OUTPUT command

The **SET OUTPUT** command allows you to define output files for the replication command line interface. The output files contain the administration SQL statements needed to set up Q replication.

Syntax

"SET OUTPUT (Q replication)" on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET OUTPUT──┬───────────────────────────┬──┬──────────────────────────┬──────────────────►
                └─CAPTURE SCRIPT "capfname"─┘  └─TARGET SCRIPT "trgfname"─┘

►──┬───────────────────────────┬──┬──────────┬──────────────────────────────────────────────►◄
   └─MONITOR SCRIPT "monfname"─┘  └─MULTIDIR─┘
```

Parameters

CAPTURE SCRIPT
Specifies the output file name for SQL scripts to be executed at the Q Capture server.

TARGET SCRIPT
Specifies the output file name for SQL scripts to be executed at the Q Apply server.

MULTIDIR
Specifies the output files to be named after the databases that the SQL script is to be run on. This keyword is optional.

MONITOR SCRIPT
Specifies the output file name for scripts to be run at the Monitor control server. The default file name is replmonitor.sql.

Usage notes

- All output will be appended to the script if a script already exists.
- Do not provide names for files for the **MULTIDIR** keyword. The ASNCLP program automatically names the output scripts based on the names of the databases that they run on.

SET LOG command

The **SET LOG** command allows you to define the log file for the replication command line interface. The log file contains informational messages, warnings, and errors.

Syntax

“SET LOG” on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET LOG──"logfilename"─────────────────────────────────────────────────────────►◄
```

Parameters

"logfilename"
: Specifies the output log file name.

Usage notes

The default log file name is qreplmsg.log.

SET TRACE command

The **SET TRACE** command allows you to enable and disable the internal trace for the ASNCLP commands.

Syntax

“SET TRACE” on page 200 has the dotted decimal version of the syntax diagram.

```
►►──SET TRACE──┬─OFF─┬─────────────────────────────────────────────────────────────►◄
               └─ON──┘
```

Parameters

OFF Turns off the trace. This is the default.

ON Turns on the trace.

Usage notes

All output is dumped to the console. For readability, you should save the output to a file.

SET RUN SCRIPT command

The **SET RUN SCRIPT** command allows you to control whether to automatically run each task command from an input file before the next task command is processed.

This command is useful for input files that contain task commands that assume that a previous command was executed and thus that the required objects already exist in the DB2 database.

Syntax

"SET RUN SCRIPT" on page 198 has the dotted decimal version of the syntax diagram.

```
►►──SET RUN SCRIPT──┬─LATER─────────────────────────┬────────────────────────────────────────────►◄
                    └─NOW STOP ON SQL ERROR──┬─ON──┬┘
                                             └─OFF─┘
```

Parameters

LATER
Specifies that the generated SQL scripts not be automatically executed.

NOW Specifies that the generated SQL scripts be automatically executed.

STOP ON SQL ERROR
Specifies whether the running of the SQL scripts should be terminated upon DB2 errors.

ON Specifies that the ASNCLP commands stop processing when the first SQL statement fails. All previous SQL statements related to this command will be rolled back. Only statements related to the current database will be rolled back.

OFF Specifies that the ASNCLP commands will execute all the SQL statements, regardless of errors.

SET CAPTURE SCHEMA command

The **SET CAPTURE SCHEMA** command allows you to set a default source Q Capture schema for all task commands. By default, the replication API uses the ASN Q Capture schema, so if you do not need additional or different Q Capture schemas, you do not need to use this command.

This command allows users of the task commands to ignore the Q Capture schema settings.

Syntax

"SET CAPTURE SCHEMA (Q replication)" on page 195 has the dotted decimal version of the syntax diagram.

```
►►──SET CAPTURE SCHEMA SOURCE──┬─TO──┬─DEFAULT─┬─┬──────────────────────────────────────►◄
                               │     └─NULLS───┘ │
                               └─capschema───────┘
```

Parameters

SOURCE
Specifies the Capture schema at the source. Can be any valid DB2 schema name.

DEFAULT
Specifies that the Q Capture schema will be set to ASN by the ASNCLP commands, that is, that any previous **SET CAPTURE SCHEMA** commands will be reset.

NULLS
Specifies that the Q Capture schema will be set to NULL.

capschema
Schema that the Q Capture control tables will be generated with.

SET APPLY SCHEMA command

"SET APPLY SCHEMA" on page 195 has the dotted decimal version of the syntax diagram.

The **SET APPLY SCHEMA** command allows you to set a default source and target Q Apply schema for all task commands.

Syntax

```
►►──SET APPLY SCHEMA──┬─TO DEFAULT──┬──────────────────────────────────────────────────►◄
                      └─applyschema─┘
```

Parameters

TO DEFAULT
Specifies that the Q Apply schema will be set to ASN by the ASNCLP commands, that is, that any previous **SET APPLY SCHEMA** commands will be reset.

applyschema
Specifies the Q Apply schema name.

SHOW SET ENV command

"SHOW SET ENV" on page 200 has the dotted decimal version of the syntax diagram.

The **SHOW SET ENV** command displays the environment set during the session. The environment is displayed to the console.

Syntax

```
►►──SHOW SET ENV───────────────────────────────────────────────────────────────────────►◄
```

Chapter 9. Environment commands for bidirectional and peer-to-peer Q replication

The Q replication environment commands specific to bidirectional and peer-to-peer replication are:

- "SET SERVER command"
- "SET MULTIDIR SCHEMA command" on page 84
- "SET SUBGROUP command" on page 84
- "SET CONNECTION command" on page 85
- "SET TABLES command" on page 85
- "SET REFERENCE TABLE command" on page 86
- "DROP SUBGROUP command" on page 87

SET SERVER command

The **SET SERVER** command allows you to specify the server (database) used in the ASNCLP session. You can specify authentication information and other required parameters for connecting to the server.

Syntax

"SET SERVER (bidirectional and peer-to-peer Q replication)" on page 199 has the dotted decimal version of the syntax diagram.

```
►►──SET SERVER──┬─CAPTURE──┬──TO──┬─NULLS────────────────────────────────────────────────────────┬────────►◄
                ├─TARGET───┤      └─┬─DB dbalias──────────┬──┬───────────────────┬──other-options─┘
                └─MULTIDIR─┘        └─DBALIAS aliasname───┘  └─DBNAME dbname─────┘
```

other-options:

```
├──┬───────────┬──┬──────────────┬────────────────────────────────────────────────────────────────┤
   └─ID userid─┘  └─PASSWORD pwd─┘
```

Parameters

CAPTURE, TARGET, MULTIDIR
Specifies the type of server being associated with the database alias:

- CAPTURE: Q Capture control server
- TARGET: Target server
- MULTIDIR: Database will act as a bidirectional or peer-to-peer replication node. For z/OS, this is the subsystem location name.

NULLS
Specifies that the server names are to be set to NULL. This option resets previously set servers.

DB, DBALIAS
Specifies the DB2 alias name (used with Linux, UNIX, or Windows).

DBNAME
Specifies the DB2 database name (used with z/OS).

ID Specifies the user ID to use when connecting to this server.

PASSWORD
Specifies the password to use when connecting to this server.

Usage notes

When using bidirectional or peer-to-peer replication, the **MULTIDIR** option must be explicitly set.

SET MULTIDIR SCHEMA command

The **SET MULTIDIR SCHEMA** command allows you to set the Q Capture and Q Apply schema to the value specified on the specified server for bidirectional or peer-to-peer replication processing. If you do not specify the server or schema, the ASNCLP defaults to ASN.

Syntax

“SET MULTIDIR SCHEMA” on page 196 has the dotted decimal version of the syntax diagram.

```
►►──SET MULTIDIR SCHEMA servername.schemaname─────────────────────────────────────────►◄
```

Parameters

servername
Name of the database that the catalogs are on.

schemaname
Q Capture and Q Apply catalog tables schema used for the node.

SET SUBGROUP command

The **SET SUBGROUP** command is used to specify the subgroup name of the bidirectional or peer-to-peer replication scenario. This is the name of the subgroup that groups the bidirectional or peer-to-peer subscriptions.

Syntax

“SET SUBGROUP” on page 200 has the dotted decimal version of the syntax diagram.

```
►►──SET SUBGROUP subgroup-name────────────────────────────────────────────────────────►◄
```

Parameters

subgroup-name
Name of the subgroup that groups the bidirectional or peer-to-peer Q subscriptions.

SET CONNECTION command

The **SET CONNECTION** command sets the connection for the direction between the two nodes.

Syntax

"SET CONNECTION" on page 195 has the dotted decimal version of the syntax diagram.

```
►►──SET CONNECTION──┬───────────────────────────────┬──────────────────────────────────────►
                    └─SUBNAME subscription-name─┘

►─SOURCE servername.schemaname─TARGET servername.schemaname──REPLQMAP queue-map-name──────────────►◄
```

Parameters

SUBNAME
Specifies the name of the Q subscriptions between the two nodes (from source to target) specified in the connection. If more than one Q subscription is created between the two nodes, the first Q subscription will carry the name as specified, and every subsequent Q subscription will have an incremental number appended to it.

SOURCE

servername
Name of the server that the source node resides on.

schemaname
Name of the replication catalog tables that the source node uses.

TARGET

servername
Name of the server that the target node resides on.

schemaname
Name of the replication catalog tables that the target node uses.

REPLQMAP
Specifies the name of the replication queue map between the Q Capture catalog of the source node and the Q Apply catalog of the target node.

SET TABLES command

The **SET TABLES** command is used to specify the source and target tables (that is, the tables are both target and source at the same time) that participate in the bidirectional or peer-to-peer replication setup.

Syntax

"SET TABLES" on page 200 has the dotted decimal version of the syntax diagram.

```
                 ┌─,──────────────────────────────────────┐
                 ▼                                        │
►►──SET TABLES(──── servername.schemaname.tableowner.tablename─┴── )──────────────────────►◄
```

Parameters

servername
Name of the server (database) that this table resides on.

schemaname
Name of the catalog tables that this table uses to replicate its data.

tableowner
Schema of the table.

tablename
Name of the table.

Usage notes

- You must specify at least one table; this table will act as both a source and a target table. If you specify additional tables and they already exist, they will be checked; if they do not exist, they will be created. If you do not specify any additional tables, the required tables will be generated automatically.
- To successfully create a set of peer-to-peer or bidirectional replication Q subscriptions using the tables specified in the **SET TABLES** command, a **CREATE QSUB** command must be issued before the next **SET TABLES** command. That is, each **SET TABLES** command will override the previous one until a **CREATE QSUB** statement is issued.

SET REFERENCE TABLE command

The **SET REFERENCE TABLE** command allows you to specify a table the participates in a group.

Syntax

“SET REFERENCE TABLE” on page 198 has the dotted decimal version of the syntax diagram.

```
►►──SET REFERENCE TABLE USING SCHEMA servername.schemaname──USES TABLE tableowner.tablename────────────►◄
```

Parameters

servername
Name of the server (database) that this table resides on.

schemaname
Name of the catalog tables that this table uses to replicate its data.

tableowner
Schema of the table.

tablename
Name of the table.

DROP SUBGROUP command

The **DROP SUBGROUP** commands deletes the subgroup.

Syntax

“DROP SUBGROUP” on page 194 has the dotted decimal version of the syntax diagram.

```
►►──DROP SUBGROUP────────────────────────────────────────────────────────────────►◄
```

Chapter 10. Control table definition commands for Q replication

The control table definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command.

The control table definition commands include:

- "CREATE CONTROL TABLES command"
- "DROP CONTROL TABLES command" on page 94

CREATE CONTROL TABLES command

The **CREATE CONTROL TABLES** command sets up Q Capture and Q Apply control tables. For event publishing, Q Apply control tables are not needed.

Restrictions: For bidirectional and peer-to-peer replication, be sure that on each server, the Q Capture and Q Apply programs use the same schema.

Syntax

"CREATE CONTROL TABLES (Q replication)" on page 182 has the dotted decimal version of the syntax diagram.

```
►►──CREATE CONTROL TABLES FOR──┬─CAPTURE SERVER USING  capparms-clause──────┬──────────────────────────►
                               └─APPLY SERVER─┬──────────────────────────┬─┘
                                              └─USING  appparms-clause──┘

►──┬────────────────────────────────┬──┬──────────────────────────────┬──────────────────────────────►◄
   └─IN──┬─ZOS── zos-ts-clause──┬─┘  └─ZOS──── INDEX── zos-idx-clause──┘
         └─UW uw-ts-clause──────┘
```

capparms-clause:

```
├──RESTARTQ─ "rstqname" ──ADMINQ─ "admqname" ──┬───────────────────────┬──┬──────────────────────┬──────►
                                               └─STARTMODE──┬─ WARMSI─┬─┘  └─MEMORY LIMIT limit──┘
                                                            ├─ COLD───┤
                                                            ├─ WARMNS─┤
                                                            └─ WARMSA─┘

►──┬──────────────────┬──┬──────────────────────────────┬──┬─────────────────────────────┬────────────►
   └─AUTOSTOP──┬─ N─┬─┘  └─MONITOR INTERVAL interval──┘  └─MONITOR LIMIT monlimit──┘
               └─ Y─┘

►──┬─────────────────────────┬──┬─────────────────────────┬──┬──────────────────────────────────┬─────►
   └─TRACE LIMIT trclimit──┘  └─SIGNAL LIMIT siglimit──┘  └─PRUNE INTERVAL prninterval──┘

►──┬──────────────────────────────────┬──┬──────────────────┬──┬───────────────────┬──┬──────────────┬─►
   └─SLEEP INTERVAL sleepinterval──┘  └─LOGREUSE──┬─ N─┬─┘  └─LOGSTDOUT──┬─ N─┬─┘  └─TERM──┬─ Y─┬─┘
                                                  └─ Y─┘                 └─ Y─┘            └─ N─┘
```

```
►─┬─────────────────────────────┬──────────────────────────────────────────────────────────────┤
  └─CAPTURE PATH "capture_path"─┘
```

apparms-clause:

```
├─┬────────────────────────┬──┬─────────────────────┬──┬────────────────────────────┬────────────►
  └─MONITOR LIMIT monlimit─┘  └─TRACE LIMIT trclimit─┘  └─MONITOR INTERVAL interval─┘

►─┬──────────────────────────────┬──┬──────────────┬──┬──────────────────────────────┬──────────►
  └─PRUNE INTERVAL prninterval─┘  └─AUTOSTOP─┬─N─┬─┘  └─SLEEP INTERVAL sleepinterval─┘
                                             └─Y─┘

►─┬──────────────────┬──┬───────────────────┬──┬──────────────────────────┬──┬──────────────┬──►
  └─LOGREUSE─┬─N─┬─┘  └─LOGSTDOUT─┬─N─┬─┘  └─APPLY PATH "apply_path"─┘  └─TERM─┬─Y─┬─┘
             └─Y─┘                └─Y─┘                                        └─N─┘

►─┬──────────────────────┬──┬────────────────────────┬──┬───────────────────────┬──────────────┤
  └─PWDFILE "filename"─┘  └─DEADLOCK RETRIES num─┘  └─SIGNAL LIMIT siglimit─┘
```

zos-ts-clause:

```
├─┬─────────────────────────────────────────────────────────────────────────┬──────────────────►
  └─PAGE LOCK─┬───────────┬─┬─tsname────────────────┬─┤ prof-clause ├─┘
              └─DB─ name─┘  └─NAMING PREFIX prefix─┘

►─┬───────────────────────────────────────────────────────────────────────┬──────────────────┤
  └─ROW LOCK─┬───────────┬─┬─tsname────────────────┬─┤ prof-clause ├─┘
             └─DB─ name─┘  └─NAMING PREFIX prefix─┘
```

uw-ts-clause:

```
├─┬───────────────────────────────────────────────────────┬──────────────────────────────────┤
  └─TBSPACE─┬─tsname────────────────┬─prof-clause─┘
            └─NAMING PREFIX prefix─┘
```

prof-clause:

```
├─┬──────────────────────────────────────┬─────────────────────────────────────────────────────┤
  └─CREATE─┬──────────────────────────┬─┘
           └─ USING PROFILE pname─┘
```

zos-idx-clause:

```
├─┬────────────────────────────────┬─────────────────────────────────────────────────────────────┤
  └─CREATE─ USING PROFILE pname─┘
```

Parameters

capparms-clause:

RESTARTQ
Name of the restart queue used by this Q Capture instance.

ADMINQ
Name of the admin queue used by this Q Capture instance. The admin queue input is the name of the administration queue on the Q Apply server to use.

STARTMODE

WARMSI
The Q Capture program will try to perform a warm start. If it is the first time that the Q Capture program runs, it will perform a cold start.

COLD The Q Capture program performs a cold start.

WARMNS
The Q Capture program attempts a warm start if information is available; if the information is not available, the Q Capture program will stop.

WARMSA
The Q Capture program will try to perform a warm start. If the capture program cannot warm start, it will always switch to a cold start and refresh the target tables.

MEMORY LIMIT
Specifies the maximum size (in megabytes) of memory that the Q Capture program can use to build transactions.

AUTOSTOP

N The Q Capture program does not terminate after retrieving the transactions until the end of the active log.

Y The Q Capture program terminates after retrieving the transactions until the end of the active log.

MONITOR INTERVAL
Specifies how frequently (in seconds) the Q Capture program inserts rows into the Q Capture monitor tables.

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

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

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

PRUNE INTERVAL
Specifies how frequently (in seconds) the Q Capture monitor tables

(IBMQREP_CAPMON and IBMQREP_CAPQMON), Q Capture trace (IBMQREP_CAPTRACE), and signal (IBMQREP_SIGNAL) tables are pruned.

SLEEP INTERVAL
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 is restarted.

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 is restarted.

LOGSTDOUT

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

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 UDB terminates. This is the default.

N The Q Capture program continues running if DB2 UDB terminates with MODE(QUIESCE).

CAPTURE_PATH
Specifies the location (absolute path, in double quotation marks for case sensitivity) of the work files used by the Q Capture program. For z/OS, it can be a MVS dataset high level qualifier with //. The default is NULL.

appparms-clause:

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

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

MONITOR INTERVAL
Specifies how frequently (in seconds) the Q Apply program inserts rows into the Q Apply monitor (IBMQREP_APPMON) table.

PRUNE INTERVAL
Specifies how frequently (in seconds) the Q Apply monitor (IBMQREP_APPMON) and Q Apply trace (IBMQREP_APPTRACE) tables are pruned.

AUTOSTOP

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

Y
: The Q Apply program terminates after all queues are emptied once. Internally, this is determined by getting a end-of-queue return code from WebSphere MQ for all the receive queues.

SLEEP INTERVAL
: Specifies the number of seconds 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 is restarted.

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 is restarted.

LOGSTDOUT

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

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

APPLY PATH
: Specifies the location (absolute path, in double quotation marks for case sensitivity) of the work files used by the Q Apply program. The default is the directory where the **asnqapp** command was invoked.

TERM

N
: The Q Apply program continues running if DB2 UDB terminates with MODE(QUIESCE).

Y
: The Q Apply program terminates if DB2 UDB terminates.

PWDFILE
: Specifies the name of the password file.

DEADLOCK RETRIES
: Specifies the number of retries for SQL deadlock errors; in addition to the database delays, the Q Apply program waits one second between each retry.

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

zos-ts-clause:

PAGE LOCK, ROW LOCK

tsname
: The table space name used for tablespace of type page lock or row lock for the z/OS control table table space.

DB
: Specifies the name of the logical database to create the table space in.

NAMING PREFIX
Specifies the prefix used to name the table space generated.

uw-ts-clause:

TBSPACE

tsname Table space name used for the table space of the Linux, UNIX, or Windows control table table space.

NAMING PREFIX
Specifies the prefix used to name the table space generated.

prof-clause:

CREATE
Specifies the table space to be created. When used on its own without the **USING PROFILE** keyword, the table space is assumed to exist and the control table is created.

USING PROFILE
Uses a particular profile to the customized table space attributes.

zos-idx-clause:

CREATE USING PROFILE
Specifies the profile name used in generating the name for the index space to be created on z/OS.

Example

Example for Q Apply:

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

Example for Q Capture:

```
CREATE CONTROL TABLES FOR CAPTURE SERVER USING RESTARTQ "restartq" ADMINQ "adminq"
STARTMODE COLD MEMORY LIMIT 4 MONITOR INTERVAL 3;
```

Example for Q Capture for z/OS:

```
CREATE CONTROL TABLES FOR CAPTURE SERVER USING
RESTARTQ "RESTARTQ"
ADMINQ "ADMINQ" CAPTURE PATH "//QCV8"
IN zos page lock DB "JQCNTLDB" QCCNTLP CREATE USING PROFILE "QCNTL_TS"
row lock DB "JQCNTLDB" QCCNTLR create using profile "QCNTL_TS";
```

Example for Q Apply for z/OS:

```
CREATE CONTROL TABLES FOR APPLY SERVER USING
APPLY PATH "//QAV8"
IN zos page lock DB "JQCNTLDB" QACNTLP CREATE USING PROFILE "QCNTL_TS"
row lock DB "JQCNTLDB" QACNTLR create using profile "QCNTL_TS";
```

DROP CONTROL TABLES command

The **DROP CONTROL TABLES** command allows you to drop the replication control catalogs on the Q Capture and Q Apply servers.

Syntax

"DROP CONTROL TABLES (Q replication)" on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP CONTROL TABLES ON──┬─CAPTURE SERVER─┬──────────────────────────────────────────────►◄
                            └─APPLY SERVER───┘
```

Parameters

CAPTURE SERVER
Drop Q Capture control tables.

APPLY SERVER
Drop Q Apply control tables.

Usage notes

This command is used in conjunction with the **SET SERVER** command to indicate where the control tables to be dropped are located.

Example

```
DROP CONTROL TABLES ON CAPTURE SERVER;
```

```
DROP CONTROL TABLES ON APPLY SERVER;
```

Chapter 11. Publishing queue map definition commands for event publishing

The publishing queue map definition commands include:

- “CREATE PUBQMAP command”
- “ALTER PUBQMAP command” on page 98
- “DROP PUBQMAP command” on page 100

CREATE PUBQMAP command

The **CREATE PUBQMAP** command allows you to set up a queue map for XML publications.

Syntax

“CREATE PUBQMAP” on page 187 has the dotted decimal version of the syntax diagram.

```
►►──CREATE PUBQMAP qmapname──┬─────────────────────────┬──USING SENDQ "sendqname" ──────────────────────►
                             └─DESC "description" ─┘

►──┬──────────────────────────┬──┬────────────────────────┬──┬───────────────────────────────┬──────────►
   └─MESSAGE CONTENT TYPE──┬─ T─┬┘  └─ERROR ACTION──┬─ I─┬─┘  └─HEARTBEAT INTERVAL interval─┘
                          └─ R─┘                  └─ S─┘

►──┬─────────────────────────┬──┬──────────────────────────┬────────────────────────────────────────►◄
   └─MAX MESSAGE SIZE size─┘  └─HEADER──┬─NONE────┬─┘
                                        └─MQ RFH2─┘
```

Parameters

qmapname
: The name of the publishing queue map.

DESC Specifies a description of publishing queue map.

SENDQ
: Specifies the name of the send queue of the WebSphere queue.

MESSAGE CONTENT TYPE
: Indicates whether messages put on the queue will contain an entire database transaction or a row operation only.

T Transaction. Messages contain all the row (update, insert, or delete) operations within a DB2 transaction, and information about the transaction. This is the default.

R Row. Messages contain a single update, insert, or delete operation, and information about the DB2 transaction to which it belongs.

ERROR ACTION
: Tells the Q Capture program what to do when the send queue is no longer accepting messages. For example, the queue might be full, or the queue manager might have reported a severe error for this queue.

I Invalidate. The Q Capture program invalidates all Q subscriptions and XML publications for this queue but continues to put messages on other queues. This is the default.

S Stop. The Q Capture program stops when an error is detected on this queue.

HEARTBEAT INTERVAL
Specifies the interval in seconds between heartbeat messages sent by the Q Capture program to a user application when there are no transactions to publish.

MAX MESSAGE SIZE
Specifies the maximum size of buffer in kilobytes used for sending messages over the send queue.

HEADER
Specifies which topic header type to use for the XML publication.

NONE
Default; only the XML publication message is sent with no special headers.

MQ RFH2
Attaches a special header from the topic column of the XML publication to the XML message.

Example

```
CREATE PUBQMAP "MyPubQMap" USING SENDQ "sendq" MESSAGE CONTENT TYPE R
ERROR ACTION I HEARTBEAT INTERVAL 5 MAX MESSAGE SIZE 4;
```

ALTER PUBQMAP command

The **ALTER PUBQMAP** command allows you to change attributes for an existing publishing queue map.

Syntax

“ALTER PUBQMAP” on page 177 has the dotted decimal version of the syntax diagram.

```
►►──ALTER PUBQMAP qmapname──USING options──────────────────────────────────────►◄
```

options:

```
├──┬──────────────────────┬──┬──────────────────────────────┬──┬─────────────────┬──────────►
   └─DESC "description"───┘  └─MESSAGE CONTENT TYPE──┬─T─┬──┘  └─SENDQ sendqname─┘
                                                     └─R─┘

►──┬─────────────────────┬──┬────────────────────────────┬──┬─────────────────────┬─────────►
   └─ERROR ACTION──┬─I─┬─┘  └─HEARTBEAT INTERVAL interval─┘  └─MAX MESSAGE SIZE size─┘
                   └─S─┘

►──┬────────────────────────┬─────────────────────────────────────────────────────────┤
   └─HEADER──┬─NONE───┬─────┘
             └─MQ RFH2┘
```

Parameters

qmapname
Name of the publishing queue map.

DESC Specifies a description of publishing queue map.

MESSAGE CONTENT TYPE
Indicates whether messages put on the queue will contain an entire database transaction or a row operation only.

T Transaction. Messages contain all the row (update, insert, or delete) operations within a DB2 transaction, and information about the transaction. This is the default.

R Row. Messages contain a single update, insert, or delete operation, and information about the DB2 transaction to which it belongs.

SENDQ *sendqname*
Updates the queues used by the publication queue map.

ERROR ACTION

I Invalidate. The Q Capture program will invalidate all subscriptions for the queue in error, but keep publishing on the other queues.

S Stop. The Q Capture program terminates when an error is detected on this queue.

HEARTBEAT INTERVAL
Specifies the interval in seconds between heartbeat messages sent by the Q Capture program to a user application when there are no transactions to publish.

MAX MESSAGE SIZE
Specifies the maximum size of buffer in kilobytes used for sending messages over the send queue.

HEADER
Specifies which topic header type to use for the XML publication.

NONE
Default; only the XML publication message is sent with no special headers.

MQ RFH2
Attaches a special header from the topic column of the XML publication to the XML message.

Example

```
ALTER PUBQMAP "MyPubQMap" USING MESSAGE CONTENT TYPE T ERROR ACTION S
HEARTBEAT INTERVAL 6 MAX MESSAGE SIZE 5;
```

DROP PUBQMAP command

The **DROP PUBQMAP** command allows you to drop an existing publishing queue map.

Restriction: The XML publications using the publishing queue map must first be deleted.

Syntax

"DROP PUBQMAP" on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP PUBQMAP qmapname──────────────────────────────────────────────────────►◄
```

Parameters

qmapname
The name of the publishing queue map to drop.

Example

```
DROP PUBQMAP "MyPubQMap";
```

Chapter 12. Replication queue map definition commands for Q replication

A replication queue map identifies the WebSphere MQ queues that a Q Capture program and a Q Apply program use to transport data and communicate. The replication queue map definition commands include:

- "CREATE REPLQMAP command"
- "ALTER REPLQMAP command" on page 102
- "DROP REPLQMAP command" on page 104

CREATE REPLQMAP command

The **CREATE REPLQMAP** command allows you to create a queue map for Q subscriptions.

Syntax

"CREATE REPLQMAP" on page 190 has the dotted decimal version of the syntax diagram.

```
►►──CREATE REPLQMAP qmapname──┬──────────────────────┬──USING ADMINQ "admnqname"──RECVQ "recvqname"──────►
                              └─DESC "description"──┘

►─SENDQ "sendqname"──┬──────────────────────┬──┬────────────────────┬──┬──────────────────────┬──────────►
                     └─NUM APPLY AGENTS num─┘  └─MEMORY LIMIT limit─┘  └─ERROR ACTION──┬─I─┬──┘
                                                                                       └─S─┘

►──┬───────────────────────────┬──┬───────────────────────┬─────────────────────────────────────────────►◄
   └─HEARTBEAT INTERVAL interval┘  └─MAX MESSAGE SIZE size─┘
```

Parameters

qmapname
The name of the replication queue map to create.

DESC Specifies a description of the replication queue map.

ADMINQ
Specifies the name of the administrative queue at the Q Capture server.

Note: If the Q Capture server does not share the same queue manager as the Q Apply server, the name of the **ADMINQ** specified is the name of the remote queue on the queue manager of the Q Apply server that sends data to the **ADMINQ** (of type Local Queue) of the Q Capture server.

RECVQ
Specifies the name of the receive queue used by the Q Apply program. The receive queue input is for the name of the receive queue on the Q Apply server to use.

SENDQ
Specifies the name of the send queue used by the Q Capture program. The

send queue input is for the name of the send queue on the Q Capture server to use. It is used for moving messages to a Q Apply program.

NUM APPLY AGENTS
Specifies the number of threads used for concurrently applying transactions for this queue.

MEMORY LIMIT
Specifies the maximum number of megabytes used per queue for buffering incoming transactions.

ERROR ACTION
Action to perform if Q Apply fails to apply the change.

I Invalidate. The Q Capture program will invalidate all Q subscriptions for the queue in error but keep publishing on the other queues.

S Stop. The Q Capture program terminates when an error is detected on this queue.

HEARTBEAT INTERVAL
Specifies the interval in seconds between heartbeat messages sent by the Q Capture program to both a user application and the Q Apply program on this queue, when there are no transactions to publish.

MAX MESSAGE SIZE
Specifies the maximum size of buffer in kilobytes used for sending messages over the send queue.

Example

```
CREATE REPLQMAP "MyRepQMap" USING ADMINQ "adminq"
RECVQ "recvqA" SENDQ "sendqA"
NUM APPLY AGENTS 3 MEMORY LIMIT 9 ERROR ACTION I
HEARTBEAT INTERVAL 5 MAX MESSAGE SIZE 4;
```

ALTER REPLQMAP command

The **ALTER REPLQMAP** command allows you customize attributes for the replication queue map.

Syntax

“ALTER REPLQMAP” on page 179 has the dotted decimal version of the syntax diagram.

```
►►──ALTER REPLQMAP qmapname──USING options──────────────────────────────────►◄
```

options:

```
├──┬──────────────────────┬──┬─────────────────────┬──┬────────────────────┬──┬────────────────────┬──►
   └─DESC "description"───┘  └─ADMINQ "admnqname"──┘  └─RECVQ "recvqname"──┘  └─SENDQ "sendqname"──┘

►──┬──────────────────────┬──┬────────────────────┬──┬─────────────────────┬──────────────────────────►
   └─NUM APPLY AGENTS num─┘  └─MEMORY LIMIT limit─┘  └─ERROR ACTION──┬─I─┬─┘
                                                                     └─S─┘
```

```
►─┬─────────────────────────────┬──┬──────────────────────┬──────────────────────────────────────────┤
  └─HEARTBEAT INTERVAL interval─┘  └─MAX MESSAGE SIZE size─┘
```

Parameters

qmapname
The name of the replication queue map to alter.

DESC Specifies a description of the replication queue map.

NUM APPLY AGENTS
Specifies the number of threads used for concurrently applying transactions for this queue.

ADMINQ
Specifies the name of the administrative queue at the Q Capture server.

Note: If the Q Capture server does not share the same queue manager as the Q Apply server, the name of the **ADMINQ** specified is the name of the remote queue on the queue manager of the Q Apply server that sends data to the Adminq (of type Local Queue) of the Q Capture server.

RECVQ
Specifies the name of the receive queue used by the Q Apply program. The receive queue input is for the name of the receive queue on the Q Apply server to use.

SENDQ
Specifies the name of the send queue used by the Q Capture program. The send queue input is for the name of the send queue on the Q Capture server to use. It is used for moving messages to a Q Apply program.

MEMORY LIMIT
Specifies the maximum number of megabytes used per queue for buffering incoming transactions.

ERROR ACTION
Action to perform if Q Apply fails to apply the change.

I Invalidate. The Q Capture program will invalidate all Q subscriptions for the queue in error but keep publishing on the other queues.

S Stop. The Q Capture program terminates when an error is detected on this queue.

HEARTBEAT INTERVAL
Specifies the interval in seconds between heartbeat messages sent by the Q Capture program to both a user application and the Q Apply program on this queue, when there are no transactions to publish.

MAX MESSAGE SIZE
Specifies the maximum size of buffer in kilobytes used for sending messages over the send queue.

Example

```
ALTER REPLQMAP "MyRepQMap" USING NUM APPLY AGENTS 4
MEMORY LIMIT 10 ERROR ACTION S
HEARTBEAT INTERVAL 4 MAX MESSAGE SIZE 5;
```

DROP REPLQMAP command

The **DROP REPLQMAP** command allows you to delete existing replication queue maps from the catalogs.

Restriction: Q subscriptions that use the replication queue map must first be deleted.

Syntax

"DROP REPLQMAP" on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP REPLQMAP qmapname──────────────────────────────────────────────────►◄
```

Parameters

qmapname
: The name of the replication queue map to drop.

Example

```
DROP REPLQMAP MyRepQMap;
```

Chapter 13. XML publication definition commands for event publishing

Each XML publication is a single object that identifies:

- The source table that you want to publish changes from
- The columns and rows from the source table that you want to be published
- The publishing queue map, which names the WebSphere MQ queue that changes are published to

The XML publication definition commands include:

- "CREATE XML PUB command"
- "ALTER XML PUB command" on page 108
- "DROP XML PUB command" on page 110

CREATE XML PUB command

The **CREATE XML PUB** command allows you to create an XML publication.

Syntax

"CREATE XML PUB" on page 191 has the dotted decimal version of the syntax diagram.

```
►►──CREATE XML PUB──┬──────────────────────────┬──────────────────────────────────────────►
                    └──USING PUBQMAP qmapname──┘


   ┌─,──────────────────────────────────────────────────────────────────────────────┐
   ▼                                                                                │
►──(──┬──────────────────┬──┬─────────────┬──┬──────────────────┬──src-clause──┬──────────────┬──)─┴──────►◄
      └─PUBNAME pubname──┘  └─DESC "desc"─┘  └─PUBQMAP name─────┘              └─opt-clause───┘
```

src-clause:

```
├──┬──┬─────────────────┬──source_name───────────────────────────────────┬──┬───────────────────┬──────┤
   │  └─source_owner.───┘                                                │  └─COLS col-cause────┘
   ├──┬──────────────────────────────────┬──┬────────────────────────────────┬──┤
   │  └─SRC OWNER LIKE── "predicate1"────┘  └─SRC NAME LIKE── "predicate2"───┘  │
   └─SRC ALL─────────────────────────────────────────────────────────────────┘
```

col-cause:

```
├──┬─ALL─────────────────────────────┬──┬──────────────────────────────┬──────────────────────┤
   │              ┌─,───────┐        │  │          ┌─,───────┐         │
   │              ▼         │        │  │          ▼         │         │
   ├─INCLUDE (──── COLNAME──┴── )────┤  └─IS_KEY (──── COLNAME──┴── )──┘
   │              ┌─,───────┐        │
   │              ▼         │        │
   └─EXCLUDE (──── COLNAME──┴── )────┘
```

opt-clause:

```
├──┬──────────────────────────────┬──┬──────────────────────────┬──┬───────────────────────┬──────────►
   └─SEARCH CONDITION "search_cond"─┘  └─ALL CHANGED ROWS──┬─N─┬──┘  └─BEFORE VALUES──┬─N─┬──┘
                                                           └─Y─┘                      └─Y─┘

►──┬──────────────────────────┬──┬───────────────────────┬──┬──────────────────────────┬──────────────►
   └─CHANGED COLS ONLY──┬─Y─┬─┘  └─HAS LOAD PHASE──┬─N─┬─┘  └─SUPPRESS DELETES──┬─N─┬─┘
                        └─N─┘                      └─E─┘                        └─Y─┘

►──┬─────────────────┬────────────────────────────────────────────────────────────────────────────────┤
   └─TOPIC "topic"───┘
```

Parameters

USING PUBQMAP
Specifies the name of the publication queue map used by all subsequent XML publications in this command.

PUBNAME
Specifies the name of the XML publication.

DESC Specifies a description of the XML publication.

PUBQMAP
Name of the publication queue map used by this publication. If the **USING PUBQMAP** keyword is not specified, then the **PUBQMAP** keyword is required for each and every publication to be defined.

src-clause:

source_owner
Schema of the source table.

source_name
Name of the source table.

SRC OWNER LIKE
Choose all tables with a schema that matches the expression in the LIKE statement. The following example show LIKE statements:

```
CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "ASN%");

CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%");
```

SRC NAME LIKE
Choose all tables with a name that matches the expression in the LIKE statement. The following example shows a LIKE statement:

```
CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC NAME LIKE "%4%")
```

SRC ALL
Choose all tables, with the exception of catalog views, that exist on the Capture server.

COLS Specifies the column options.

opt-clause:

SEARCH CONDITION

Specifies a search condition for filtering changes to replicate. The change is not sent if the predicate is false. This is an annotated select WHERE clause, where the column names of the table to replicated must be enclosed in colons (see example below). The following example shows a WHERE clause:

```
CREATE XML PUB USING PUBQ MAP ASNMAP
(PUBNAME mypubname ALLTYPE1 OPTIONS SEARCH CONDITION
"WHERE :MYKEY: > 1000");
```

ALL CHANGED ROWS

Data sending option.

Y Yes; send a row when any column has changed for the source table.

N No; send a row only if the subscribed column has changed for the source table.

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 No. 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 Yes. When there are changes to non-key columns in the source table that are part of a Q subscription or XML publication, 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 Q subscription or XML publication only if they have changed. This field applies to update operations only.

Y Yes. When the Q Capture program sends an updated row, it sends only the changed columns that are part of a Q subscription or XML publication. This is the default.

N No. The Q Capture program sends all columns in a row that are part of a Q subscription or XML publication whenever any of them has changed.

HAS LOAD PHASE

Indicates whether the target table for the Q subscription or XML 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. 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.

In the case of target type stored procedure the **I** option is not valid.

SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table.

N No. Do not send deleted rows. This is the default.

Y Yes. Send deleted rows.

col-clause:

ALL Publish all columns on the source table.

INCLUDE
Specifies columns to be published. You can specify multiple columns.

EXCLUDE
Specifies columns to be excluded from publication. You can specify multiple columns.

IS_KEY
Indicates whether the column is part of the key to be used for replication or publishing. Any set of columns that are unique at the source can be used.

0 The column is not part of the unique key. Its order in the transaction message will be the same as its order in the source table. This is the default.

col_order
The column is part of the unique key. In a multiple-column key, the column's order in the transaction message will be encoded based on the *col_order* that you specify.

At least one of the columns from the source table should have a value greater than 0 in the IBMQREP_SRC_COLS table or the Q subscription or XML publication will be invalid.

TOPIC
The **TOPIC** specified will be attached as a header and used by any XML publication which has the MQ RFH2 tag specified in its publication queue map.

Example

```
CREATE XML PUB USING PUBQMAP MyPubQMap
(PUBNAME "MyXMLPub" ERIC.TSTTABLE
ALL CHANGED ROWS Y BEFORE VALUES Y CHANGED COLS ONLY Y
HAS LOAD PHASE N SUPPRESS DELETES Y);
```

ALTER XML PUB command

The **ALTER XML PUB** command allows you to alter an XML publication.

Syntax

"ALTER XML PUB" on page 180 has the dotted decimal version of the syntax diagram.

```
►►──ALTER XML 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─┬───┘
                                                             └─Y─┘                      └─Y─┘

►─┬────────────────────────────┬──┬───────────────────────────┬──┬─────────────────┬──────────────────────┤
  └─CHANGED COLS ONLY──┬─Y─┬───┘  └─SUPPRESS DELETES──┬─N─┬───┘  └─TOPIC "topic"───┘
                       └─N─┘                          └─Y─┘
```

Parameters

pubname
: Name of the XML publication.

source_name
: Name of the source table.

source_owner
: Schema of the source table.

DESC Specifies a description of the XML publication.

PUBQMAP
: Specifies the name of the publication queue map used by this XML publication.

other-opt-clause:

SEARCH CONDITION
: Used for filtering rows for replication. The change is not sent if the predicate is false. This is an annotated select WHERE clause. The column names of the source table are enclosed in colons (see example below). The following example shows a WHERE clause:

```
ALTER XML PUB mypubname FOR ALLTYPE1 OPTIONS
SEARCH CONDITION "WHERE :MYKEY: > 1000";
```

ALL CHANGED ROWS
: Data sending option.

Y Yes; send a row when any column has changed for the source table.

N No; send a row only if the subscribed column has changed for the source table.

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 No. 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.

Y Yes. When there are changes to non-key columns in the source table that are part of a Q subscription or XML publication, 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 Q subscription or XML publication only if they have changed. This field applies to update operations only.

Y Yes. When the Q Capture program sends an updated row, it sends only the changed columns that are part of a Q subscription or XML publication. This is the default.

N No. The Q Capture program sends all columns in a row that are part of a Q subscription or XML publication whenever any of them has changed.

SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table.

N No. Do not send deleted rows. This is the default.

Y Yes. Send deleted rows.

TOPIC
The **TOPIC** specified will be attached as a header and used by any XML publication which has the MQ RFH2 tag specified in its publication queue map.

Example

```
ALTER XML PUB "MyXMLPub" FOR ERIC.TSTTABLE OPTIONS ALL CHANGED ROWS N
BEFORE VALUES N CHANGED COLS ONLY N HAS LOAD PHASE E SUPPRESS DELETES N;
```

DROP XML PUB command

The **DROP XML PUB** command allows you to drop an XML publication.

Syntax

"DROP XML PUB" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──DROP XML PUB──┬─────┬──┬────────────────────────────────┬──────────────────────────────────►◄
                  └─ALL─┘  │  ┌─,─────────────────────┐     │
                           │  ▼                       │     │
                           ├───(─PUBNAME── pubname─)──┴─────┤
                           └─FOR PUBNAME LIKE "predicate"───┘
```

Parameters

ALL This keyword will drop all the XML publications for the schema and server set through the SET commands.

PUBNAME
Name of the XML publication to be dropped.

FOR PUBNAME LIKE
Drop all XML publications that match the LIKE statement. The following example shows a LIKE statement:

```
DROP XML PUB FOR PUBNAME LIKE "pubname02%";
```

Example

```
DROP XML PUB (PUBNAME "MyXMLPub");
```

Chapter 14. Q subscription definition commands for Q replication

The Q subscription definition commands include:

- “CREATE QSUB command (unidirectional replication)”
- “ALTER QSUB command” on page 119
- “DROP QSUB command” on page 121

CREATE QSUB command (unidirectional replication)

The **CREATE QSUB** command allows you to create a Q subscription

Syntax

“CREATE QSUB (unidirectional Q replication)” on page 189 has the dotted decimal version of the syntax diagram.

```
►►──CREATE QSUB──┬───────────────┬──┬──────────────────────────┬──────────────────────────────────────────►
                 └─SUBTYPE─ U────┘  └─USING REPLQMAP── name────┘


   ┌─,──────────────────────────────────────────────────────────────────────────────────────────┐
   ▼                                                                                            │
►──(──┬─────────────────────┬──┬────────────────┬──┬──────────────────────┬──src-clause──┬────────────────┬──)─┴──────────►◄
      └─SUBNAME── subname───┘  └─DESC── "desc"──┘  └─REPLQMAP── name──────┘              └─trg-clause─────┘
```

src-clause:

```
├──┬──┬──────────────────┬──source_name───────────────────────────────────────┬──┬───────────────────────┬──┤
   │  └─source_owner.────┘                                                    │  └─OPTIONS opt-clause────┘
   ├──┬─────────────────────────────────┬──┬────────────────────────────────┬─┤
   │  └─SRC OWNER LIKE── "predicate1"───┘  └─SRC NAME LIKE── "predicate2"───┘ │
   └─SRC ALL──────────────────────────────────────────────────────────────────┘
```

opt-clause:

```
├──┬──────────────────────────────────────────┬──┬──────────────────────────┬──┬───────────────────────────┬──►
   └─SEARCH CONDITION "search_condition"──────┘  └─ALL CHANGED ROWS──┬─ N─┬──┘  └─HAS LOAD PHASE──┬─ N─┬────┘
                                                                     └─ Y─┘                      ├─ I─┤
                                                                                                 └─ E─┘

►──┬─────────────────────────────┬──────────────────────────────────────────────────────────────────────────┤
   └─SUPPRESS DELETES──┬─ N─┬────┘
                       └─ Y─┘
```

trg-clause:

```
├──┬─────────┬──┬──────────────────────────────────────────────────────┬──────────────────────────────────────►
   └─EXIST───┘  └─TARGET──┬─NAME──┬──────────────┬──target_name─────┬──┘
                          │       └─owner──.─────┘                  │
                          └─NAMING PREFIXprefix─────────────────────┘
```

```
►─┬──────────────────────────────────────────────────────────┬──┬─────────────────────────┬──────────────────────►
  └─IN─┬─────────┬──┬─ tsname──────────────┬──┬──────────────┬┘  └─TYPE──┬─USERTABLE──┬─┘
       └─DB name─┘  └─NAMING PREFIX prefix─┘  └─prof-clause──┘           ├─STOREDPROC─┤
                                                                         └─NICKNAME───┘

►─┬──────────────────────────────────────────────────────┬─────────────────────────────────────────────────────────►
  └─TRGCOLS──┬─ALL───────────────────────────────────────┤
             │             ┌─,──────────────────────┐    │
             │             ▼                        │    │
             ├─INCLUDE (───trgcolname──┬────────────┬┴─ )─┤
             │                         └─srccolname─┘    │
             │             ┌─,────────────┐              │
             │             ▼              │              │
             └─EXCLUDE (─── trgcolname────┴─ )───────────┘

►─┬─────────────────────────────────────────────────────────────────────┬──────────────────────────────────────────►
  │                                              ┌─,─────────────┐      │
  │                                              ▼               │      │
  └─KEYS──┬──────────────────────────┬── (─── keyname──┬─────┬───┴─ )──┘
          ├─ indexowner.indexname────┤                 ├─ + ─┤
          └─NAMING PREFIX prefix─────┘                 └─ - ─┘

►─┬──────────────────────────────────────────┬─────────────────────────────────────────────────────────────────────►
  └─ZOS INDEX CREATE USING PROFILE pname─────┘

►─┬──────────────────────────────────────────────────────────────────────────┬──┬────────────────────────┬───────►
  ├─NICKNAME──┬─ owner.nickname──────────────────────────────────────────────┤  └─CONFLICT ACTION──┬─ I─┬┘
  │           └─NAMING PREFIX prefix─┘                                       │                    ├─ F─┤
  └─NEW NICKNAME RMT SERVERNAME── srvname──┬─ owner.nickname─────────────────┘                    ├─ D─┤
                                           └─NAMING PREFIX prefix─┘                               ├─ S─┤
                                                                                                  └─ Q─┘

►─┬──────────────────────┬──┬────────────────────────────┬──┬────────────────────┬──────────────────────────────►◄
  └─ERROR ACTION──┬─ S─┬─┘  └─OKSQLSTATES "sqlstates" ───┘  └─LOAD TYPE──┬─ 0─┬─┘
                  ├─ D─┤                                                 ├─ 1─┤
                  └─ Q─┘                                                 ├─ 2─┤
                                                                         └─ 3─┘
```

prof-clause:

```
├──┬────────────────────────────────────┬──────────────────────────────────────────────────────────────────────────┤
   └─CREATE──┬──────────────────────────┬┘
             └─USING PROFILE pname──────┘
```

Parameters

SUBTYPE

U Unidirectional replication.

USING REPLQMAP
Specifies the name of the global replication queue map to be used by all the Q subscriptions in this task command. This is the replication queue map that will be used by all the Q subscriptions in a mass scenario, or if replication queue maps are not specified with the parenthesis for each Q subscription.

SUBNAME
Specifies the name of the Q subscription.

DESC Specifies a description of the Q subscription.

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

src-clause:

source_owner
Schema of the source table.

source_name
Name of the source table.

SRC OWNER LIKE
Choose all tables with a schema that matches the expression in the LIKE statement. The following example shows a LIKE statement:

```
CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "ASN%");

CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%");
```

SRC NAME LIKE
Choose all tables with a name that matches the expression in the LIKE statement. The following example shows a LIKE statement:

```
CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "ASN%");

CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%");
```

SRC ALL
Choose all tables, with the exception of catalog views, that exist on the Q Capture server.

trg-clause:

EXIST Specifies that the target table exists.

- If you specify **EXIST** but do not provide a target table name, ASNCLP will look for the default table, which will be ″TGT-<SOURCE TABLE NAME>″.
- If you specify **EXIST** and a single **TARGET NAME**, and you use **SOURCE ALL** or **SOURCE NAME LIKE**, then all 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 using the default named ″TGT<sourcetablename>″.

TARGET NAME

target_owner
Schema of the target table.

target_name
Name of the target table.

NAMING PREFIX
Specifies the prefix to use for 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 then be called CLPT1.

IN

DB 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.

NAMING PREFIX
Prefix to be used in the name of the table space.

TYPE

USERTABLE
Target is a table.

STOREDPROC
Target is a stored procedure.

TRGCOLS

ALL Replicate all columns from the source table.

INCLUDE
Specifies the column definitions if the target table *target_owner.target_name* does not exist. Specifies the columns in the target table, if it exists, to participate in the replication.

trgcolname
Add a column definition to the target table using 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, three) );
```

srccolname
Add a column definition to the target table using the provided name and use the properties of the source column for the target column properties.

EXCLUDE
Exclude the source column from the target table definition. The following example shows how you would use the **CREATE QSUB** command with this option. In this example, the source table columns are one, two, and three and the target table columns are een, twee, and drie.

```
CREATE QSUB USING REPLQMAP replqmap10
( SUBNAME sub10 dpropr64.srctable exist target name
dpropr64.tgttable trgcols
exclude(een one,twee two,drie three));
```

trgcolname
Exclude a column definition to the target table using the provided name and the properties of a source column with the same name.

KEYS

indexowner
Owner of the index to be created.

indexname
Name of the index to be created.

NAMING PREFIX
Specifies the prefix of the index to be created.

keyname
Name of the columns to be included in the index.

+ Ascending order.

- Descending order.

ZOS INDEX CREATE USING PROFILE
Specifies the name of the index profile for customizing z/OS index.

NICKNAME
Specifies the nickname to be used by the Q Apply program to load rows to the target table.

owner Owner of the source nickname.

nickname
Name of the source nickname.

NEW NICKNAME RMT SERVERNAME
Specifies the name of the remote server if the nickname is to be created by the ASNCLP program.

NAMING PREFIX
Specifies the prefix to be given to the nickname when the ASNCLP program generates it.

CONFLICT ACTION

I Ignore.

F Force; this requires the send option **CHANGED COLS ONLY = 'N'**.

D Disable the Q subscription.

S Stop Q Apply.

Q Stop reading from queue.

ERROR ACTION

S Stop Q Apply without applying the transaction at all.

D Disable subscription and notify Q Capture.

Q Stop reading from queue.

OKSQLSTATES
List of SQLSTATES within double quotation marks that are not to be considered as error when applying changes to this table.

LOAD TYPE

0 Choose the best type automatically.

1 Use crossloader only.

2 Use export/import only.

3 Use export/load only.

prof-clause:

CREATE
Specifies that the table space is to be created.

USING PROFILE
Specifies the name of the profile to be used when creating the table space.

opt-clause:

SEARCH CONDITION
Specifies a search condition for filtering changes to replicate. The change is not sent if the predicate is false. This is an annotated select WHERE clause, where the column names of the table to be replicated must be enclosed in colons (see example below). The following example shows a WHERE clause:

```
CREATE XML PUB USING PUBQ MAP ASNMAP
(PUBNAME mypubname ALLTYPE1 OPTIONS SEARCH CONDITION
"WHERE :MYKEY: > 1000");
```

ALL CHANGED ROWS
Data sending option.

N No; send a row only if the subscribed column has changed for the source table.

Y Yes; send a row when any column has changed for the source table.

HAS LOAD PHASE
Indicates whether the target table for the Q subscription or XML publication will be loaded with data from the source.

N No load phase at the target. This is the default.

I Internal load. Specifies an automatic load. The Q Apply program calls the LOAD from CURSOR utility, EXPORT/IMPORT utility, or EXPORT/LOAD utility, depending on the LOAD_TYPE specified in the IBMQREP_TARGETS table, and on the platform of the Q Apply server and Q Capture server.

E External load. 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.

In the case of target type stored procedure the **I** option is not valid.

SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table.

N No. Do not send deleted rows. This is the default.

Y Yes. Send deleted rows.

Usage notes

- If a target table is specified, but SRC ALL or SRC NAME LIKE was specified, all the source tables will attempt to subscribe to target tables with the same name.
- DB value for Logical Database is mandatory for target tables on z/OS platforms. It must be specified in the profile.
- If mass subscription is used (for example, using the SRC OWNER LIKE or SRC NAME LIKE clause) the specified *target_owner target_name* clause is not valid if the target table does not exist. Only default or naming prefix are allowed for generated target tables.

Example

```
CREATE QSUB USING REPLQMAP MyRepQMap
(SUBNAME "MyQSub" ERIC.TSTTABLE OPTIONS ALL CHANGED ROWS Y
HAS LOAD PHASE N SUPPRESS DELETES Y
TARGET NAME ERIC.TSTTRGT TYPE USERTABLE CONFLICT ACTION I
ERROR ACTION S LOAD TYPE 0);
```

ALTER QSUB command

The **ALTER QSUB** command allows you to alter a Q subscription.

Syntax

"ALTER QSUB (unidirectional Q replication)" on page 178 has the dotted decimal version of the syntax diagram.

```
►►──ALTER QSUB subname──REPLQMAP mapname──┬──────────────────────────┬──┬────────────────────┬──────────►
                                          └─USING REPLQMAP mapname───┘  └─DESC description───┘

►──┬────────────────────────────────┬─────────────────────────────────────────────────────────────────►◄
   └─USING OPTIONS  other-opt-clause┘
```

other-opt-clause:

```
├──┬─────────────────────────────────────┬──┬──────────────────────────┬──┬─────────────────────────┬──►
   └─SEARCH CONDITION "search_condition"─┘  └─ALL CHANGED ROWS──┬─N─┬──┘  └─HAS LOAD PHASE──┬─N─┬──┘
                                                                └─Y─┘                       ├─I─┤
                                                                                            └─E─┘

►──┬─────────────────────────┬──┬─────────────────────────┬──┬──────────────────────┬───────────────────►
   └─SUPPRESS DELETES──┬─N─┬─┘  └─CONFLICT ACTION──┬─I─┬──┘  └─ERROR ACTION──┬─S─┬──┘
                       └─Y─┘                       ├─F─┤                     ├─D─┤
                                                   ├─D─┤                     └─Q─┘
                                                   ├─S─┤
                                                   └─Q─┘

►──┬───────────────────────────┬──┬──────────────────┬──────────────────────────────────────────────────┤
   └─OKSQLSTATES "sqlstates"───┘  └─LOAD TYPE──┬─0─┬─┘
                                               ├─1─┤
                                               ├─2─┤
                                               └─3─┘
```

Parameters

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

USING REPLQMAP
The Q subscription can be altered to use another replication queue map.

DESC Specifies a description of the Q subscription.

other-opt-clause:

SEARCH CONDITION
Specifies a search condition for filtering changes to replicate. The change is not sent if the predicate is false. This is an annotated select WHERE clause,

where the column names of the table to be replicated must be enclosed in colons (see example below). The following example shows a WHERE clause:

```
CREATE XML PUB USING PUBQ MAP ASNMAP
(PUBNAME mypubname ALLTYPE1 OPTIONS SEARCH CONDITION
"WHERE :MYKEY: > 1000");
```

ALL CHANGED ROWS
Data sending option.

N No; send a row only if the subscribed column has changed for the source table.

Y Yes; send a row when any column has changed for the source table.

HAS LOAD PHASE
Indicates whether the target table for the Q subscription or XML publication will be loaded with data from the source.

N No load phase at the target. This is the default.

I Internal load. Specifies an automatic load. The Q Apply program calls the LOAD from CURSOR utility, EXPORT/IMPORT utility, or EXPORT/LOAD utility, depending on the LOAD_TYPE specified in the IBMQREP_TARGETS table, and on the platform of the Q Apply server and Q Capture server.

E External load. 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.

In the case of target type stored procedure the **I** option is not valid.

SUPPRESS DELETES
Specifies whether to send rows that were deleted from the source table.

N No. Do not send deleted rows. This is the default.

Y Yes. Send deleted rows.

CONFLICT ACTION

I Ignore.

F Force; this requires the send option **CHANGED COLS ONLY = ′N′**.

D Disable subscription.

S Stop Q Apply.

Q Stop reading from queue.

ERROR ACTION

S Stop the Q Apply program without applying the transaction at all.

D Disable subscription and notify the Q Capture program.

Q Stop reading from queue.

OKSQLSTATES
List of SQLSTATES within double quotation marks that are not to be considered as error when applying changes to this table.

LOAD TYPE

0 Choose the best type automatically.

1 Use crossloader only.

2 Use export/import only.

3 Use export/load only.

ADD COLUMN

colname
Name of a single column on the source table of the Q subscription to be added to the list of subscribed columns in the Q subscription.

Usage notes

If the ADD COLUMN option is specified, no other alter attributes are allowed.

Example

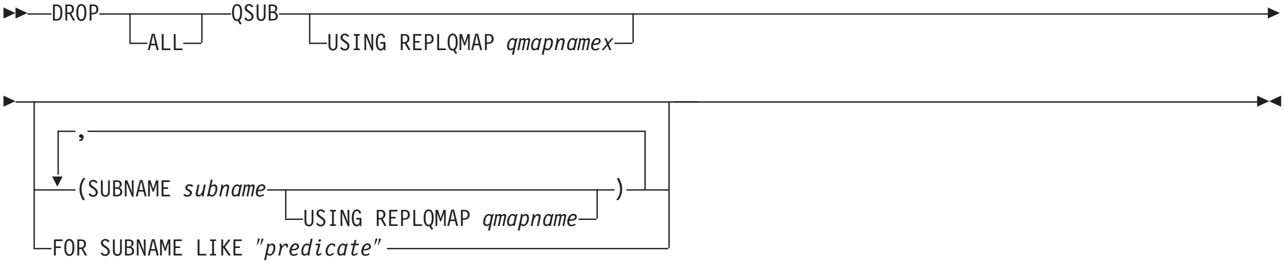
```
ALTER QSUB "MyQSub" REPLQMAP MyRepQMap
USING OPTIONS ALL CHANGED ROWS N HAS LOAD PHASE I
SUPPRESS DELETES N CONFLICT ACTION F ERROR ACTION Q LOAD TYPE 1;
```

DROP QSUB command

The **DROP QSUB** command drops an existing subscription set for a specified Apply qualifier.

Syntax

"DROP QSUB" on page 193 has the dotted decimal version of the syntax diagram.



Parameters

ALL Drops all Q subscriptions.

USING REPLQMAP
Drop all Q subscriptions using the particular replication queue map.

SUBNAME
Specifies the name of the Q subscription to be dropped.

USING REPLQMAP
Specifies the name of the replication queue map between the Q Capture catalog of the source node and the Q Apply catalog of the target node.

FOR SUBNAME LIKE
Drop all Q subscriptions that match the expression in the LIKE statement. The following example shows a LIKE statement:

```
CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "ASN%");

CREATE XML PUB USING PUBQMAP ABCDPUBQMAP
(SRC OWNER LIKE "JDOE" SRC NAME LIKE "%TAB%");
```

Example

```
DROP QSUB (SUBNAME "MyQSub" USING REPLQMAP MyRepQMap);
```

Chapter 15. Bidirectional or peer-to-peer replication script command for Q replication

This command is required for setting up peer-to-peer and bidirectional replication scenarios. All commands for generating a bidirectional or peer-to-peer replication setup must be prewritten to an input file and loaded into ASNCLP using the **LOAD MULTIDIR REPL SCRIPT** command.

LOAD MULTIDIR REPL SCRIPT command

The **LOAD MULTIDIR REPL SCRIPT** command is used to invoke ASNCLP scripts used to set up peer-to-peer and bidirectional replication.

Syntax

“LOAD MULTIDIR REPL SCRIPT” on page 194 has the dotted decimal version of the syntax diagram.

```
►►──LOAD MULTIDIR REPL SCRIPT "filelocation/filename"──────────────────────────────────────►◄
```

Parameters

filelocation
: Absolute path where the input file is located. If no directory is specified, the current directory is assumed.

filename
: Name of the bidirectional or peer-to-peer replication input file.

Usage notes

- Only definitions pertaining to one subgroup can be placed into 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 input file.

Example

The following is a sample ASNCLP 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 ASNCLP script creates four subgroups. Each of the subgroup’s definitions 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 the information required to generate the subgroup’s Q subscriptions are located in this one input file.

Chapter 16. Bidirectional replication definition commands for Q replication

The bidirectional replication definition commands are:

- “CREATE QSUB command (bidirectional replication)”
- “ALTER QSUB command (bidirectional replication)” on page 127
- “DROP SUBTYPE command (bidirectional replication)” on page 129

CREATE QSUB command (bidirectional replication)

The **CREATE QSUB** command allows you to create a Q subscription for bidirectional replication.

Syntax

“CREATE QSUB (bidirectional Q replication)” on page 188 has the dotted decimal version of the syntax diagram.

```
►►──CREATE QSUB SUBTYPE B──────────────────────────────────────────────────────────────►

►──┬──────────────────────────────────────────────────────────────────────────┬─────────►◄
   └─FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause─┘


►►──┬──────────────────────────────────────────────────────────────────────────┬────────►◄
    └─FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause─┘
```

source-clause:

```
├──┬──────────────────────────┬──┬───────────────────────────┬──────────────────────────┤
   └─ALL CHANGED ROWS──┬─ N─┬─┘  └─HAS LOAD PHASE──┬─ N─┬─┘
                       └─ Y─┘                      ├─ I─┤
                                                   └─ E─┘
```

target-clause:

```
├──┬──────────────────────────┬──┬──────────────────────────┬──┬──────────────────────────┬──►
   └─CONFLICT RULE──┬─ K─┬─┘  └─CONFLICT ACTION──┬─ I─┬─┘  └─ERROR ACTION──┬─ Q─┬─┘
                    ├─ C─┤                       ├─ F─┤                    ├─ D─┤
                    └─ A─┘                       ├─ D─┤                    └─ S─┘
                                                 ├─ S─┤
                                                 └─ Q─┘

►──┬─────────────────────┬──┬─────────────────────────────┬──────────────────────────────┤
   └─LOAD TYPE──┬─ 0─┬─┘  └─OKSQLSTATES "sqlstates" ─┘
                ├─ 2─┤
                └─ 3─┘
```

Parameters

SUBTYPE B
Specifies bidirectional Q subscriptions.

source-clause:

ALL CHANGED ROWS
Data sending option.

N No; send a row only if the subscribed column has changed for the source table.

Y Yes; send a row when any column has changed for the source table.

HAS LOAD PHASE
Indicates whether the target table for the Q subscription or XML publication will be loaded with data from the source.

N No load phase at the target. This is the default.

I Internal load. Specifies an automatic load. The Q Apply program calls the LOAD from CURSOR utility, EXPORT/IMPORT utility, or EXPORT/LOAD utility, depending on the LOAD_TYPE specified in the IBMQREP_TARGETS table, and on the platform of the Q Apply server and Q Capture server.

E External load. 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.

In the case of target type stored procedure the **I** option is not valid.

target-clause:

CONFLICT RULE

K Check only key values.

C Check changed non-key values in addition to key values.

A Check all values for updates.

CONFLICT ACTION

I Ignore.

F Force; this requires the send option **SUBSCRIPTIONS CHANGED COLS ONLY = ’N’**.

D Disable the Q subscription.

S Stop the Q Apply program.

Q Stop reading from the queue.

ERROR ACTION

Q Stop reading from the queue.

D Disable the Q subscription and notify the Q Capture program.

S Stop the Q Apply program without applying the transaction at all.

OKSQLSTATES
List of SQLSTATES within double quotation marks that are not to be considered as error when applying changes to this table.

LOAD TYPE

0 Choose the best type automatically.

2 Use export/import only.

3 Use export/load only.

Usage notes

For conflict handling rules, only the listed combinations are permitted for bidirectional replication.

Note: None of the **Before Values** or **Change Cols Only** attributes can be set explicitly using the administration tool.

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

ALTER QSUB command (bidirectional replication)

The **ALTER QSUB** command allows you to alter a Q subscription for bidirectional replication.

Syntax

“ALTER QSUB (bidirectional Q replication)” on page 177 has the dotted decimal version of the syntax diagram.

```
►►──ALTER QSUB SUBTYPE B──────────────────────────────────────────────────────────►

►─┬─────────────────────────────────┬─┬─────────────────────┬─┬─────────────────────┬──────────►◄
  └─FROM NODE servername.schemaname─┘ └─SOURCE source-clause─┘ └─TARGET target-clause─┘


►►──┬─────────────────────────────────┬─┬─────────────────────┬─┬─────────────────────┬────────►◄
    └─FROM NODE servername.schemaname─┘ └─SOURCE source-clause─┘ └─TARGET target-clause─┘
```

source-clause:

```
├──┬────────────────────────┬─┬────────────────────────┬──────────────────────────────────────┤
   └─ALL CHANGED ROWS─┬─N─┬─┘ └─HAS LOAD PHASE─┬─N─┬─┘
                      └─Y─┘                    ├─I─┤
                                               └─E─┘
```

target-clause:

```
├──┬─────────────────────────┬──┬────────────────────────────┬──┬────────────────────────┬──────────►
   └─CONFLICT RULE──┬─ K─┬─┘  └─CONFLICT ACTION──┬─ I─┬─┘  └─ERROR ACTION──┬─ Q─┬─┘
                    ├─ C─┤                       ├─ F─┤                    ├─ D─┤
                    └─ A─┘                       ├─ D─┤                    └─ S─┘
                                                 ├─ S─┤
                                                 └─ Q─┘

►──┬─────────────────────┬──┬──────────────────────────────┬──────────────────────────────────┤
   └─LOAD TYPE──┬─ 0─┬─┘  └─OKSQLSTATES "sqlstates"─┘
                ├─ 2─┤
                └─ 3─┘
```

Parameters

SUBTYPE B
Specifies bidirectional Q subscriptions.

source-clause:

ALL CHANGED ROWS
Data sending option.

N No; send a row only if the subscribed column has changed for the source table.

Y Yes; send a row when any column has changed for the source table.

HAS LOAD PHASE
Indicates whether the target table for the Q subscription or XML publication will be loaded with data from the source.

N No load phase at the target. This is the default.

I Internal load. Specifies an automatic load. The Q Apply program calls the LOAD from CURSOR utility, EXPORT/IMPORT utility, or EXPORT/LOAD utility, depending on the LOAD_TYPE specified in the IBMQREP_TARGETS table, and on the platform of the Q Apply server and Q Capture server.

E External load. 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.

In the case of target type stored procedure the **I** option is not valid.

target-clause:

CONFLICT RULE

K Check only key values.

C Check changed non-key values in addition to key values.

A Check all values for updates.

CONFLICT ACTION

I Ignore.

F Force; this requires the send option **SUBSCRIPTIONS CHANGED COLS ONLY = 'N'**.

D Disable Q subscription.

S Stop Q Apply.

Q Stop reading from the queue.

ERROR ACTION

Q Stop reading from the queue.

D Disable Q subscription and notify the Q Capture program.

S Stop the Q Apply program without applying the transaction at all.

OKSQLSTATES
List of SQLSTATES within double quotation marks that are not to be considered as error when applying changes to this table.

LOAD TYPE

0 Choose the best type automatically.

2 Use export/import only.

3 Use export/load only.

DROP SUBTYPE command (bidirectional replication)

The **DROP SUBTYPE** command allows you to drop a Q subscription for bidirectional replication.

Syntax

"DROP SUBTYPE (bidirectional Q replication)" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──DROP SUBTYPE── B──QSUBS──────────────────────────────────────────────────────►◄
```

Parameters

SUBTYPE B
Specifies bidirectional Q subscription.

Usage notes

No tables or table spaces are ever dropped.

Chapter 17. Peer-to-peer replication definition commands for Q replication

The peer-to-peer replication definition commands include:

- "CREATE QSUB command (peer-to-peer replication)"
- "ALTER QSUB command (peer-to-peer replication)" on page 132
- "DROP SUBTYPE command (peer-to-peer replication)" on page 134

CREATE QSUB command (peer-to-peer replication)

The **CREATE QSUB** command creates a Q subscription for peer-to-peer replication.

Syntax

"DROP SUBTYPE (bidirectional Q replication)" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──CREATE QSUB──SUBTYPE P──┬──────────────────────┬──┬──────────────────────┬──────────────►◄
                            └─SOURCE source-clause─┘  └─TARGET target-clause─┘
```

source-clause:

```
├──┬────────────────────────┬──────────────────────────────────────────────────────────────┤
   └─HAS LOAD PHASE──┬─ N─┬─┘
                     ├─ I─┤
                     └─ E─┘
```

target-clause:

```
├──┬──────────────────────┬──┬─────────────────────┬──┬─────────────────────────┬──────────┤
   └─ERROR ACTION──┬─ Q─┬─┘  └─LOAD TYPE──┬─ 0─┬──┘  └─OKSQLSTATES "sqlstates"─┘
                   ├─ D─┤                 ├─ 2─┤
                   └─ S─┘                 └─ 3─┘
```

Parameters

SUBTYPE P
Specifies peer-to-peer Q subscriptions.

source-clause:

HAS LOAD PHASE
Indicates whether or not the table for the subscription will be loaded at the target.

N No; load does not need to be performed on the tables involved in this subscription before replicating changes.

I Internal; the Q Apply program will perform the load.

E External; the Q subscription has a load phase that is carried externally by the user application.

target-clause:

ERROR ACTION
Action to perform if Q Apply fails to apply the change.

Q Stop reading from the queue.

D Disable Q subscription and notify the Q Capture program.

S Stop the Q Apply program without applying the transaction at all.

LOAD TYPE

0 Choose the best type automatically.

2 Use export/import only.

3 Use export/load only.

OKSQLSTATES
List of SQLSTATES 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 bidirectional or peer-to-peer replication with convergence, only the attributes shown in Table 9 are allowed (and are implicitly assigned).

Table 9. Attributes for bidirectional or peer-to-peer replication with convergence

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

Example

```
create qsub subtype p;

create qsub subtype p SOURCE HAS LOAD PHASE N
TARGET ERROR ACTION Q LOAD TYPE 0;

create qsub subtype p source has load phase E
target error action D load type 2 OKSQLSTATES "000101";
```

ALTER QSUB command (peer-to-peer replication)

The **ALTER QSUB** command modifies an existing Q subscription for peer-to-peer replication.

Syntax

“ALTER QSUB (peer-to-peer Q replication)” on page 178 has the dotted decimal version of the syntax diagram.

```
►►─ALTER QSUB─SUBTYPE P──SOURCE source-clause─┬──────────────────────┬──────────────────►◄
                                              └─TARGET target-clause─┘
```

source-clause:

```
├──┬─────────────────────────┬──────────────────────────────────────────────────────────┤
   └─HAS LOAD PHASE──┬─ N─┬─┘
                     ├─ I─┤
                     └─ E─┘
```

target-clause:

```
├──┬───────────────────────┬──┬──────────────────┬──┬──────────────────────────┬─────────┤
   └─ERROR ACTION──┬─ Q─┬─┘  └─LOAD TYPE──┬─ 0─┬─┘  └─OKSQLSTATES "sqlstates"─┘
                   ├─ D─┤                 ├─ 2─┤
                   └─ S─┘                 └─ 3─┘
```

Parameters

SUBTYPE P
Specifies peer-to-peer Q subscriptions.

source-clause:

HAS LOAD PHASE
Indicates whether the target table for the Q subscription or XML publication will be loaded with data from the source.

N No load phase at the target. This is the default.

I Internal load. Specifies an automatic load. The Q Apply program calls the LOAD from CURSOR utility, EXPORT/IMPORT utility, or EXPORT/LOAD utility, depending on the LOAD_TYPE specified in the IBMQREP_TARGETS table, and on the platform of the Q Apply server and Q Capture server.

E External load. 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.

In the case of target type stored procedure the **I** option is not valid.

target-clause:

ERROR ACTION

D Disable Q subscription and notify the Q Capture program.

S Stop the Q Apply program without applying the transaction at all.

Q Stop reading from the queue.

LOAD TYPE

0 Choose the best type automatically.

2 Use export/import only.

3 Use export/load only.

OKSQLSTATES
List of SQLSTATES within double quotation marks that are not to be considered as error when applying changes to this table.

DROP SUBTYPE command (peer-to-peer replication)

The **DROP SUBTYPE** command deletes a Q subscription for peer-to-peer replication.

Syntax

"DROP SUBTYPE (peer-to-peer Q replication)" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──DROP SUBTYPE P QSUBS──────────────────────────────────────────────────►◄
```

Parameters

SUBTYPE P
Specifies 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.

Chapter 18. Q subscription operation commands for Q replication

The Q subscription operation commands are:

- "START QSUB command"
- "STOP QSUB command"
- "ALTER ADD COLUMN command" on page 136
- "LOAD DONE command" on page 137

START QSUB command

The **START QSUB** command allows you to start a Q subscription.

Syntax

"START QSUB" on page 200 has the dotted decimal version of the syntax diagram.

```
►►──START QSUB──┬─SUBNAME subname─────────┬──────────────────────────────────────────────►◄
                └─FOR SUBNAME LIKE "%text%"─┘
```

Parameters

SUBNAME *subname*
: Specifies the name of the Q subscription to be started.

FOR SUBNAME LIKE
: Start Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
START QSUB SUBNAME ABCDSUBNAME (SRC NAME LIKE "%4%")
```

STOP QSUB command

The **STOP QSUB** command allows you to stop a Q subscription.

Syntax

"STOP QSUB" on page 201 has the dotted decimal version of the syntax diagram.

```
►►──STOP QSUB──┬─SUBNAME subname─────────┬───────────────────────────────────────────────►◄
               └─FOR SUBNAME LIKE "%text%"─┘
```

Parameters

SUBNAME *subname*
: Specifies the name of the Q subscription to be stopped.

FOR SUBNAME LIKE
: Stop Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
STOP QSUB SUBNAME ABCDSUBNAME (SRC NAME LIKE "%4%")
```

ALTER ADD COLUMN command

The **ALTER ADD COLUMN** command allows you to add a column to a Q subscription.

Syntax

"ALTER ADD COLUMN" on page 174 has the dotted decimal version of the syntax diagram.

```
                                     ┌─,─────────┐
►►──ALTER ADD COLUMN USING SIGNAL(───▼─ colname ─┴── )──┬─QSUB subname─USING REPQMAPqmapname─┬──────────►◄
                                                         └─XML PUB pubname────────────────────┘
```

Parameters

colname
One or more columns (separated by a comma) to be added to the definition of the active Q subscription or XML publication.

QSUB

subname
Name of the Q subscription.

repqmapname
Name of the replication queue map used by the Q subscription.

XML PUB

pubname
Name of the XML publication.

Usage notes

- This command adds a column to the subscribed set of columns on the source table for an active Q subscription or XML publication.
- The column needs to exist in the source table already and should not be part of any existing subscription.
- The Q subscription or XML 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.

LOAD DONE command

The **LOAD DONE** command causes the API to insert a LOADDONE signal into the IBMQREP_SIGNAL table. This allows you to inform Q Capture that you have finished loading the target table and that Q Apply can start applying spilled changes to the target table. You need to insert the LOADDONE signal only if you are doing an external load. If Q Apply is doing the load, this signal is not necessary.

Syntax

"LOAD DONE" on page 194 has the dotted decimal version of the syntax diagram.

```
►►──LOAD DONE QSUB──┬─SUBNAME subname──────────┬──────────────────────────────────────────────►◄
                    └─FOR SUBNAME LIKE "%text%"─┘
```

Parameters

SUBNAME *subname*
Specifies the name of the Q subscription for the LOADDONE signal.

FOR SUBNAME LIKE
Start Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
LOAD DONE QSUB SUBNAME ABCDSUBNAME (SRC NAME LIKE "%4%")
```

Chapter 19. XML publication operation commands for event publishing

The event publishing operation commands are:

- “START XML PUB command”
- “STOP XML PUB command”

START XML PUB command

The **START XML PUB** command allows you to start an XML publication.

Syntax

“START XML PUB” on page 200 has the dotted decimal version of the syntax diagram.

```
►►──START─XML PUB──┬─PUBNAME pubname──────────┬──────────────────────────────────────────────►◄
                   └─FOR PUBNAME LIKE "%text%"─┘
```

Parameters

pubname
Specifies the name of the XML publication to be started.

FOR PUBNAME LIKE
Start XML publications that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
START XML PUB PUBNAME ABCDPUBNAME (SRC NAME LIKE "%4%")
```

STOP XML PUB command

The **STOP XML PUB** command allows you to stop an XML publication.

Syntax

“STOP XML PUB” on page 201 has the dotted decimal version of the syntax diagram.

```
►►──STOP─XML PUB──┬─PUBNAME pubname──────────┬───────────────────────────────────────────────►◄
                  └─FOR PUBNAME LIKE "%text%"─┘
```

Parameters

pubname
Specifies the name of the Q subscription to be stopped.

FOR PUBNAME LIKE
Stop Q subscriptions that match the expression in the LIKE clause. The following example shows a LIKE clause:

```
STOP XML PUB PUBNAME ABCDPUBNAME (SRC NAME LIKE "%4%")
```

Part 4. Replication Alert Monitor commands

Chapter 20. General Replication Alert Monitor commands

The Replication Alert Monitor definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command. Use the control table definition commands to create or drop the Replication Alert Monitor control tables.

The general Replication Alert Monitor program commands for both SQL replication and Q replication include:

- "CREATE CONTACT command"
- "ALTER CONTACT command" on page 144
- "DROP CONTACT command" on page 144
- "SUBSTITUTE CONTACT command" on page 145
- "DELEGATE CONTACT command" on page 145
- "CREATE GROUP command" on page 146
- "ALTER GROUP command" on page 146
- "DROP GROUP command" on page 147
- "SET SERVER command" on page 148

CREATE CONTACT command

The **CREATE CONTACT** command creates contact information, such as the contact name and e-mail address, that the Replication Alert Monitor program uses for notifications when a replication alert condition is detected. You can optionally associate a contact to a pre-existing group.

Syntax

"CREATE CONTACT" on page 182 has the dotted decimal version of the syntax diagram.

```
      ┌─,────────────────────────────────────────────────────────────┐
      ▼                                                              │
►►────CREATE CONTACT───contact-name──┬──────────────────────────┬────┴──┬─EMAIL─┬── "email-address" ──────────►
                                     └─GROUP── group-name──────┘        └─PAGE──┘

►──┬──────────────────────────┬────────────────────────────────────────────────────────────────────────────►◄
   └─DESCRIPTION── "desc"─────┘
```

Parameters

CONTACT *contact-name*
: Specifies the name of the contact. This name cannot match another contact already defined. This parameter is required.

GROUP *group-name*
: Specifies the name of the group. 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 *"desc"*
: Specifies a brief description for the contact. The double quotation marks are required.

ALTER CONTACT command

The **ALTER CONTACT** command alters 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" on page 176 has the dotted decimal version of the syntax diagram.

```
►►──ALTER CONTACT────contact-name──────┬─EMAIL─┬──── "email-address" ──┬──────────────────────────┬──────────►◄
                                       └─PAGE──┘                       └─DESCRIPTION── "desc" ─┘
```

Parameters

CONTACT *contact-name*
: Specifies the name of the contact. The contact must exist. This parameter is required.

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 *"desc"*
: Specifies a brief description for the contact. The double quotation marks are required.

DROP CONTACT command

The **DROP CONTACT** command drops an existing contact.

Syntax

"DROP CONTACT" on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP CONTACT────contact-name1──────┬───────────────────────────────────┬──────────────────────────►◄
                                       └─SUBSTITUTE WITH── contact-name2─┘
```

Parameters

CONTACT *contact-name1*
Specifies the name of the contact. The contact must exist. This parameter is required.

SUBSTITUTE WITH *contact-name2*
Specifies the name of the 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.

SUBSTITUTE CONTACT command

The **SUBSTITUTE CONTACT** command substitutes one existing contact with another existing contact.

Syntax

“SUBSTITUTE CONTACT” on page 201 has the dotted decimal version of the syntax diagram.

```
►►──SUBSTITUTE CONTACT────contact-name1──── WITH── contact-name2──────────────────────────────────►◄
```

Parameters

contact-name1
Specifies the name of the contact to be substituted. The contact must exist. This parameter is required.

WITH *contact-name2*
Specifies the new contact for all alert conditions (if any) that refer to the contact being substituted. The new contact must exist.

DELEGATE CONTACT command

The **DELEGATE CONTACT** command delegates an existing contact to a new contact for a specific period of time.

Syntax

“DELEGATE CONTACT” on page 192 has the dotted decimal version of the syntax diagram.

```
►►──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. This parameter is required.

TO *contact-name2*

Specifies the new contact for all alert conditions (if any) that refer to the contact being delegated. The contact must exist. This parameter is required.

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"*

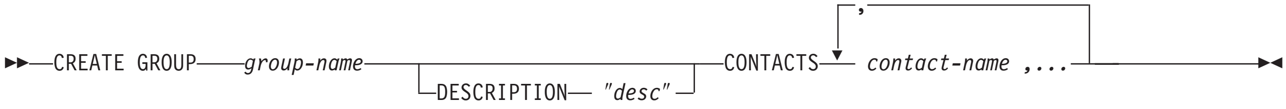
Specifies the date when the delegation ends. The date is sensitive to the DB2 locale. The double quotation marks are required.

CREATE GROUP command

The **CREATE GROUP** command creates a group of replication monitor contacts.

Syntax

"CREATE GROUP" on page 184 has the dotted decimal version of the syntax diagram.



Parameters

group-name

Specifies the name of the group. This name cannot match another group already defined. This parameter is required.

DESCRIPTION *"desc"*

Specifies a brief description for the group. The double quotation marks are required.

CONTACTS *contact-name, ...*

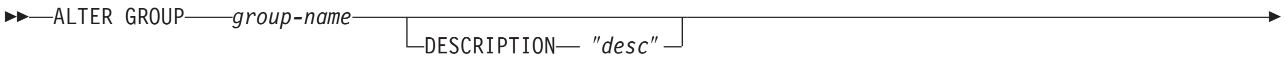
Specifies a comma-separated list of contacts that belong to this group.

ALTER GROUP command

The **ALTER GROUP** command alters a group of replication monitor contacts.

Syntax

"ALTER GROUP" on page 176 has the dotted decimal version of the syntax diagram.



```
                 ┌─,─────────────┐
                 ▼               │
►──┬─NEW CONTACTS─── contact-name1─┴──────┬──────────────────────────────────────────►◄
   │           ┌─,─────────────────────┐   │
   │           ▼                       │   │
   └─CONTACTS─── contact-name2───┬─ADD────┬┴───┘
                                 └─REMOVE─┘
```

Parameters

group-name
: Specifies the name of the group. The group must exist. This parameter is required.

DESCRIPTION *"desc"*
: 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.

DROP GROUP command

The **DROP GROUP** command drops a group of replication monitor contacts.

Syntax

"DROP GROUP" on page 193 has the dotted decimal version of the syntax diagram.

```
►►──DROP GROUP────group-name──────────────────────────────────────────────────►◄
```

Parameters

group-name
: Specifies the name of the group. The group must exist. This parameter is required.

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.

SET SERVER command

The **SET SERVER** command allows you to specify the server (database) used in the ASNCLP session. You can specify authentication information and other required parameters for connecting to the server.

You should always set the Monitor control server before running the monitor administration commands.

Syntax

"SET SERVER (Replication Alert Monitor)" on page 199 has the dotted decimal version of the syntax diagram.

```
►►──SET SERVER MONITOR──TO──┬─NULLS──────────────────────────────────────────────────┬──────────────►◄
                            └─┬─DB dbalias──────────┬──┬──────────────────┬─other-options─┘
                              └─DBALIAS aliasname───┘  └─DBNAME dbname────┘
```

other-options:

```
├──┬─────────────┬──┬──────────────────┬──────────────────────────────────────────────────────────────┤
   └─ID userid───┘  └─PASSWORD pwd─────┘
```

Parameters

MONITOR
Specifies that a monitor is associated with the database alias.

NULLS
Specifies that the server names are to be set to NULL. This option resets previously set servers.

DB, DBALIAS
Specifies the DB2 alias name (used with Linux, UNIX, or Windows).

DBNAME
Specifies the DB2 database name (used with z/OS).

ID Specifies the user ID to use when connecting to this server.

PASSWORD
Specifies the password to use when connecting to this server.

Chapter 21. Replication Alert Monitor definition commands for SQL replication

The Replication Alert Monitor definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command. Use the control table definition commands to create or drop the Replication Alert Monitor definition commands control tables.

The Replication Alert Monitor definition commands for SQL replication include:

- “CREATE ALERT CONDITIONS FOR CAPTURE command”
- “ALTER ALERT CONDITIONS FOR CAPTURE command” on page 151
- “DROP ALERT CONDITIONS FOR CAPTURE command” on page 152
- “CREATE ALERT CONDITIONS FOR APPLY command” on page 153
- “ALTER ALERT CONDITIONS FOR APPLY command” on page 155
- “DROP ALERT CONDITIONS FOR APPLY command” on page 157

CREATE ALERT CONDITIONS FOR CAPTURE command

The **CREATE ALERT CONDITIONS FOR CAPTURE** command creates 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 of contacts.

Syntax

“CREATE ALERT CONDITIONS FOR CAPTURE” on page 181 has the dotted decimal version of the syntax diagram.

```
►►──CREATE ALERT CONDITIONS FOR CAPTURE──┬─────────────────────┬──MONITOR QUALIFIER── mon-qual──────────►
                                         └─SCHEMA cap-schema───┘

                                          ┌─,──────────────────────────────┐
                                          ▼                                │
►─NOTIFY──┬─CONTACT── contact-name──┬── (───┬─STATUS DOWN──────────────────┬─┴── )─────────────────────►◄
          └─GROUP── group-name──────┘       ├─STATUS LAST COMMIT time-secs─┤
                                            ├─ERRORS───────────────────────┤
                                            ├─WARNINGS─────────────────────┤
                                            ├─CURRENT LATENCY latency──────┤
                                            ├─HISTORIC LATENCY latency─────┤
                                            └─MEMORY memory────────────────┘
```

Parameters

SCHEMA
Specifies the Capture schema that qualifies the Capture tables to be monitored. The default is ASN.

MONITOR QUALIFIER
Specifies the Monitor qualifier.

NOTIFY

Specifies the contact or group of contacts to be notified when the alert condition occurs.

CONTACT

Specifies the contact to be notified.

GROUP

Specifies the group to be notified.

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

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 using **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

Specifies that the Monitor program calculates the current latency using the values of the CURR_COMMIT_TIME and MAX_COMMIT_TIME columns in the IBMSNAP_RESTART table. If the latency is greater than the number of seconds specified, an alert is sent.

HISTORIC LATENCY

Specifies that the Monitor program calculates the current latency using the values of the MONITOR_TIME and SYNCHTIME columns in the IBMSNAP_CAPMON table. If the latency is greater than the number of seconds specified, an alert is sent.

MEMORY

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 command issues an error.

ALTER ALERT CONDITIONS FOR CAPTURE command

The **ALTER ALERT CONDITIONS FOR CAPTURE** command alters alert conditions for the Capture program.

Syntax

"ALTER ALERT CONDITIONS FOR CAPTURE" on page 175 has the dotted decimal version of the syntax diagram.

```
►►──ALTER ALERT CONDITIONS FOR CAPTURE──┬─────────────────────┬──MONITOR-QUALIFIER── mon-qual──────────►
                                        └─SCHEMA cap-schema───┘

                                                 ┌─,─────────────────────────────────────────┐
                                                 ▼                                           │
►──┬──────────────────────────────────────┬──( ──┬─ADD────┬──┬─STATUS DOWN──────────────────┬─┴── )──────►◄
   └─NOTIFY──┬─CONTACT── contact-name──┬──┘      ├─REMOVE─┤  ├─STATUS LAST COMMIT time-secs─┤
             └─GROUP── group-name──────┘         └─CHANGE─┘  ├─ERRORS───────────────────────┤
                                                             ├─WARNINGS─────────────────────┤
                                                             ├─CURRENT LATENCY latency──────┤
                                                             ├─HISTORIC LATENCY latency─────┤
                                                             └─MEMORY memory────────────────┘
```

Parameters

SCHEMA
Specifies the Capture schema that qualifies the Capture tables to be monitored. The default is ASN.

MONITOR QUALIFIER
Specifies the Monitor qualifier.

NOTIFY
Specifies the contact or group of contacts to be notified when the alert condition occurs.

CONTACT
Specifies the contact to be notified.

GROUP
Specifies the group to be notified.

ADD Specifies that you are adding an alert condition.

REMOVE
Specifies that you are removing an alert condition.

CHANGE
Specifies that you are changing 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
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 using **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 have been logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of ’ERROR’ for the OPERATION. If any row is fetched, the DESCRIPTION column is included in the alert.

WARNINGS
Specifies that the Monitor program checks if any warnings have been logged in the IBMSNAP_CAPTRACE table, specifically, any rows that have a value of ’WARNING’ for the OPERATION. If any row is fetched, the DESCRIPTION column is included in the alert.

CURRENT LATENCY
Specifies that the Monitor program calculates the current latency using the values of the CURR_COMMIT_TIME and MAX_COMMIT_TIME columns in the IBMSNAP_RESTART table. If the latency is greater than the number of seconds specified, an alert is sent.

HISTORIC LATENCY
Specifies that the Monitor program calculates the current latency using the values of the MONITOR_TIME and SYNCHTIME columns in the IBMSNAP_CAPMON table. If the latency is greater than the number of seconds specified, an alert is sent.

MEMORY
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 has 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 command issues an error.

DROP ALERT CONDITIONS FOR CAPTURE command

The **DROP ALERT CONDITIONS FOR CAPTURE** command drops alert conditions for the Capture program.

Syntax

“DROP ALERT CONDITIONS FOR CAPTURE” on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP ALERT CONDITIONS FOR CAPTURE──SCHEMA cap-schema──MONTITOR QUALIFIER mon-qual──────────────────►◄
```

Parameters

SCHEMA
Specifies the Capture schema that qualifies the Capture tables to be monitored. This parameter is required.

MONITOR QUALIFIER
Specifies the Monitor qualifier. This parameter is required.

CREATE ALERT CONDITIONS FOR APPLY command

The **CREATE ALERT CONDITIONS FOR APPLY** command creates 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 of contacts.

Syntax

"CREATE ALERT CONDITIONS FOR APPLY" on page 180 has the dotted decimal version of the syntax diagram.

```
►►──CREATE ALERT CONDITIONS FOR APPLY──QUALIFIER qual-name──┬─────────────────────┬──────────────────────►
                                                            └─SET NAME set-name───┘

►─MONITOR-QUALIFIER── mon-qual──NOTIFY──┬─CONTACT── contact-name─┬────────────────────────────────────►◄
                                        └─GROUP── group-name─────┘
```

```
►►──(──┬─STATUS DOWN──────────────┬─ )──────────────────────────────────────────────────────────────►◄
       ├─ERRORS───────────────────┤
       ├─WARNINGS─────────────────┤
       ├─SUBSCRIPTIONS FAILING────┤
       ├─SUBSCRIPTIONS DELAYED time┤
       ├─SUBSCRIPTIONS INACTIVE───┤
       ├─SUBSCRIPTIONS REFRESHED──┤
       ├─TRANSACTIONS REJECTED────┤
       ├─REWORKED ROWS rows───────┤
       └─LATENCY── end-end-latency┘
```

Parameters

APPLY QUALIFIER
Specifies the Apply qualifier.

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
Specifies the Monitor qualifier.

NOTIFY
Specifies the contact or group of contacts to be notified when the alert condition occurs.

CONTACT
Specifies the contact to be notified.

GROUP
Specifies the group to be notified.

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
Specifies whether the Monitor program checks if subscription sets were processed too late. The determination is based on the following formula: (LAST_RUN + user threshold in seconds > CURRENT TIMESTAMP).

SUBSCRIPTIONS INACTIVE
Specifies whether the Monitor program looks for subscription sets made inactive by the Apply program. Such sets are identified by a value of 0 for the ACTIVATE column and -1 for the STATUS column of the IBMSNAP_SUBS_SET table.

SUBSCRIPTIONS REFRESHED
Specifies whether the Monitor programs checks if a full refresh has been processed since the last Monitor cycle. See the FULL_REFRESH column in the IBMSNAP_APPLYTRAIL table for this information (rows from the IBMSNAP_APPLYTRAIL table whose values for FULL_REFRESH are 'Y'). If any row is fetched, an alert is sent.

TRANSACTIONS REJECTED
Specifies that the Monitor program checks if any conflict has been detected by the Apply program when updating the source table and the replica tables. This check is valid only for subscriptions in an update-anywhere replication environment. See the IBMSNAP_APPLYTRAIL table for this information. If any row is fetched, an alert is sent.

REWORKED ROWS
Specifies whether the Monitor program checks if any rows were inserted into the IBMSNAP_APPLYTRAIL table since the last Monitor cycle for rows reworked in the target table. If the number of rows fetched exceeds the specified value, an alert is sent.

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 command issues an error.

ALTER ALERT CONDITIONS FOR APPLY command

The **ALTER ALERT CONDITIONS FOR APPLY** command alters alert conditions for the Apply program.

Syntax

"ALTER ALERT CONDITIONS FOR APPLY" on page 175 has the dotted decimal version of the syntax diagram.

```
►►──ALTER ALERT CONDITIONS FOR APPLY──QUALIFIER qual-name──┬──────────────────────┬──────────────────►
                                                           └─SET NAME set-name─┘

►─MONITOR-QUALIFIER── mon-qual──NOTIFY──┬─CONTACT── contact-name─┬──────────────────────────────────►◄
                                        └─GROUP── group-name─────┘


►►──(──┬─ADD────┬──┬─STATUS DOWN──────────────────┬── )──────────────────────────────────────────────►◄
       ├─REMOVE─┤  ├─ERRORS───────────────────────┤
       └─CHANGE─┘  ├─WARNINGS─────────────────────┤
                   ├─SUBSCRIPTIONS FAILING────────┤
                   ├─SUBSCRIPTIONS DELAYED time───┤
                   ├─SUBSCRIPTIONS INACTIVE───────┤
                   ├─SUBSCRIPTIONS REFRESHED──────┤
                   ├─TRANSACTIONS REJECTED────────┤
                   ├─REWORKED ROWS rows───────────┤
                   └─LATENCY end-end-latency──────┘
```

Parameters

APPLY QUALIFIER
Specifies the Apply qualifier.

SET NAME
Specifies the subscription set name.

MONITOR QUALIFIER
Specifies the Monitor qualifier.

NOTIFY
Specifies the contact or group of contacts to be notified when the alert condition occurs.

CONTACT
Specifies the contact to be notified.

GROUP
Specifies the group to be notified.

ADD Specifies that you are adding an alert condition.

REMOVE
Specifies that you are removing an alert condition.

CHANGE
Specifies that you are changing 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

Specifies whether the Monitor program checks if subscription sets were processed too late. The determination is based on the following formula: (LAST_RUN + user threshold in seconds > CURRENT TIMESTAMP).

SUBSCRIPTIONS INACTIVE

Specifies whether the Monitor program looks for subscription sets made inactive by the Apply program. Such sets are identified by a value of 0 for the ACTIVATE column and -1 for the STATUS column of the IBMSNAP_SUBS_SET table.

SUBSCRIPTIONS REFRESHED

Specifies whether the Monitor programs checks if a full refresh has been processed since the last Monitor cycle. See the FULL_REFRESH column in the IBMSNAP_APPLYTRAIL table for this information (rows from the IBMSNAP_APPLYTRAIL table whose values for FULL_REFRESH are ’Y’). If any row is fetched, an alert is sent.

TRANSACTIONS REJECTED

Specifies that the Monitor program checks if any conflict has been detected by the Apply program when updating the source table and the replica tables. This check is valid only for subscriptions in an update-anywhere replication environment. See the IBMSNAP_APPLYTRAIL table for this information. If any row is fetched, an alert is sent.

REWORKED ROWS

Specifies whether the Monitor program checks if any rows were inserted into the IBMSNAP_APPLYTRAIL table since the last Monitor cycle for rows reworked in the target table. If the number of rows fetched exceeds the specified value, an alert is sent.

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 command issues an error.

DROP ALERT CONDITIONS FOR APPLY command

The **DROP ALERT CONDITIONS FOR APPLY**command drops alert conditions for the Apply program.

Syntax

“DROP ALERT CONDITIONS FOR APPLY” on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP ALERT CONDITIONS FOR APPLY QUALIFIER── apply-qual──MONTITOR QUALIFIER── mon-qual──────────────►◄
```

Parameters

APPLY QUALIFIER
Specifies the Apply qualifier. This parameter is required.

MONITOR QUALIFIER
Specifies the Monitor qualifier. This parameter is required.

Chapter 22. Replication Alert Monitor definition commands for Q replication

The Replication Alert Monitor definition commands are task commands that execute within the context of the replication command-line interface. They inherit the context defined by the **SET SERVER** command. Use the control table definition commands to create or drop the Replication Alert Monitor definition commands control tables.

The Replication Alert Monitor definition commands for Q replication include:

- "CREATE ALERT CONDITIONS FOR QCAPTURE command"
- "ALTER ALERT CONDITIONS FOR QCAPTURE command" on page 161
- "DROP ALERT CONDITIONS FOR QCAPTURE command" on page 162
- "CREATE ALERT CONDITIONS FOR QAPPLY command" on page 163
- "ALTER ALERT CONDITIONS FOR QAPPLY command" on page 164
- "DROP ALERT CONDITIONS FOR QAPPLY command" on page 166

CREATE ALERT CONDITIONS FOR QCAPTURE command

The **CREATE ALERT CONDITIONS FOR QCAPTURE** command allows you to create alert conditions for the Q Capture program.

Syntax

"CREATE ALERT CONDITIONS FOR QCAPTURE" on page 181 has the dotted decimal version of the syntax diagram.

```
►►──CREATE ALERT CONDITIONS FOR QCAPTURE──┬─────────────────┬──MONITOR QUALIFIER monitor-qualifier──────────►
                                          └─SCHEMA schema───┘


                                    ┌─,─────────────────────────────┐
►──notification-list-definition──(──▼─┬─STATUS DOWN─────────────────┬─┴─ )──────────────────────────────►◄
                                      ├─ERRORS──────────────────────┤
                                      ├─WARNINGS────────────────────┤
                                      ├─LATENCY── seconds───────────┤
                                      ├─MEMORY── megabytes──────────┤
                                      ├─TRANSACTION SIZE── megabytes┤
                                      ├─SUBSCRIPTIONS INACTIVE──────┤
                                      └─QUEUE DEPTH── queue_percent─┘
```

notification-list-definition:

```
├──NOTIFY──┬─CONTACT contact-name─┬─────────────────────────────────────────────────────────────────────┤
           └─GROUP group-name─────┘
```

Parameters

SCHEMA
Specifies the Q Capture schema that qualifies the process to be monitored. The default is ASN.

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.

LATENCY
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
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
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.

notification-list-definition:

CONTACT
Specifies the contact to be notified when defined alert conditions are detected.

GROUP
Specifies the group to be notified when defined alert conditions are detected.

Usage notes

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for information on how to define contacts and groups.

ALTER ALERT CONDITIONS FOR QCAPTURE command

"ALTER ALERT CONDITIONS FOR QCAPTURE" on page 176 has the dotted decimal version of the syntax diagram.

The **ALTER ALERT CONDITIONS FOR QCAPTURE** command allows you to alter the alert conditions for the Q Capture program.

Syntax

```
►►──ALTER ALERT CONDITIONS FOR QCAPTURE──┬─────────────────┬──MONITOR-QUALIFIER monitor-qualifier──────────►
                                         └─SCHEMA schema───┘


                                            ┌─,──────────────────────────────────────────┐
►──┬──────────────────────────────┬──(──────▼─┬─ADD────┬──┬─ STATUS DOWN──────────────────┬─┴─ )──────────►◄
   └─notification-list-definition─┘          ├─REMOVE─┤  ├─ERRORS────────────────────────┤
                                             └─CHANGE─┘  ├─WARNINGS──────────────────────┤
                                                         ├─LATENCY── seconds─────────────┤
                                                         ├─MEMORY── megabytes────────────┤
                                                         ├─TRANSACTION SIZE── megabytes──┤
                                                         ├─SUBSCRIPTIONS INACTIVE────────┤
                                                         └─QUEUE DEPTH── queue-percent───┘
```

notification-list-definition:

```
├──NOTIFY──┬─CONTACT contact-name─┬──────────────────────────────────────────────────────────────┤
           └─GROUP group-name─────┘
```

Parameters

SCHEMA
Specifies the Q Capture schema that qualifies the process to be monitored. The default is ASN.

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.

LATENCY
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
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 specified number of megabytes.

TRANSACTION SIZE
: 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
: Specifies that an alert will be sent when the specified percentage of the given queue is full.

notification-list-definition:

CONTACT
: Specifies the contact to be notified when defined alert conditions are detected.

GROUP
: Specifies the group to be notified when defined alert conditions are detected.

Usage notes

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for information on how to define contacts and groups.

DROP ALERT CONDITIONS FOR QCAPTURE command

The **DROP ALERT CONDITIONS FOR QCAPTURE** command allows you to drop alert conditions for the Q Capture program.

Syntax

"DROP ALERT CONDITIONS FOR QCAPTURE" on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP ALERT CONDITIONS FOR QCAPTURE SCHEMA schema──MONITOR QUALIFIER monitor-qualifier──────────────►◄
```

Parameters

SCHEMA
: Specifies the Q Capture schema that qualifies the process to be monitored. The default is ASN.

MONITOR QUALIFIER
: Specifies the monitor qualifier that groups the alert conditions.

Usage notes

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for information on how to define contacts and groups.

CREATE ALERT CONDITIONS FOR QAPPLY command

The **CREATE ALERT CONDITIONS FOR QAPPLY** command allows you to create alert conditions for the Q Apply program.

Syntax

“CREATE ALERT CONDITIONS FOR QAPPLY” on page 181 has the dotted decimal version of the syntax diagram.

```
►►──CREATE ALERT CONDITIONS FOR QAPPLY──┬─────────────────┬──MONITOR-QUALIFIER monitor-qualifier──────────►
                                        └─SCHEMA schema───┘


                                            ┌─,───────────────────────────────┐
                                            ▼                                 │
►──┬──────────────────────────────┬──(──┬── STATUS DOWN──────────────────┬──┴─ )──────────────────────────►◄
   └─notification-list-definition─┘     ├─ERRORS─────────────────────────┤
                                        ├─WARNINGS───────────────────────┤
                                        ├─LATENCY── seconds──────────────┤
                                        ├─EELATENCY── seconds────────────┤
                                        ├─MEMORY── megabytes─────────────┤
                                        ├─EXCEPTIONS─────────────────────┤
                                        ├─SPILL QUEUES DEPTH── percentage┤
                                        └─QUEUE DEPTH── queue-percent────┘
```

notification-list-definition:

```
├──NOTIFY──┬─CONTACT contact-name─┬───────────────────────────────────────────────────────┤
           └─GROUP group-name─────┘
```

Parameters

SCHEMA
Specifies the Q Apply schema that qualifies the process to be monitored. The default is ASN.

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 Apply program is down.

ERRORS
Specifies that the Monitor program check if error messages were logged in the IBMQREP_APPLYTRACE table.

LATENCY
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
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

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

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

Specifies that an alert will be sent when the specified percentage of the given queue is full.

notification-list-definition:

CONTACT

Specifies the contact to be notified when defined alert conditions are detected.

GROUP

Specifies the group to be notified when defined alert conditions are detected.

Usage notes

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for information on how to define contacts and groups.

ALTER ALERT CONDITIONS FOR QAPPLY command

The **ALTER ALERT CONDITIONS FOR QAPPLY** command allows you to alter alert conditions for the Q Apply program.

Syntax

“ALTER ALERT CONDITIONS FOR QAPPLY” on page 175 has the dotted decimal version of the syntax diagram.

```
►►──ALTER ALERT CONDITIONS FOR QAPPLY──┬─────────────────┬──MONITOR-QUALIFIER monitor-qualifier──────────►
                                       └─SCHEMA schema──┘
```

```
                                            ┌─,──────────────────────────────────────────┐
                                            ▼                                            │
►─┬──────────────────────────────┬──(──┬─ADD────┬─┬─ STATUS DOWN──────────────────────┬─┴─ )──────────►◄
  └─notification-list-definition─┘     ├─REMOVE─┤ ├─ERRORS───────────────────────────┤
                                       └─CHANGE─┘ ├─WARNINGS─────────────────────────┤
                                                  ├─LATENCY── seconds────────────────┤
                                                  ├─EELATENCY── seconds──────────────┤
                                                  ├─MEMORY── megabytes───────────────┤
                                                  ├─EXCEPTIONS───────────────────────┤
                                                  ├─SPILL QUEUES DEPTH── percentage──┤
                                                  └─QUEUE DEPTH── queue-percent──────┘
```

notification-list-definition:

```
├──NOTIFY──┬─CONTACT contact-name─┬──────────────────────────────────────────────────────────────┤
           └─GROUP group-name─────┘
```

Parameters

SCHEMA
: Specifies the Q Apply schema that qualifies the process to be monitored. The default is ASN.

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 Apply program is down.

ERRORS
: Specifies that the Monitor program check if error messages were logged in the IBMQREP_APPLYTRACE table.

LATENCY
: 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
: 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
: 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
: 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
Specifies that an alert will be sent when the specified percentage of the given queue is full.

notification-list-definition:

CONTACT
Specifies the contact to be notified when defined alert conditions are detected.

GROUP
Specifies the group to be notified when defined alert conditions are detected.

Usage notes

See the *DB2 Information Integrator Replication and Event Publishing Guide and Reference* for information on how to define contacts and groups.

DROP ALERT CONDITIONS FOR QAPPLY command

The **DROP ALERT CONDITIONS FOR QAPPLY** command allows you to drop alert conditions for the Q Apply program.

Syntax

“DROP ALERT CONDITIONS FOR QAPPLY” on page 192 has the dotted decimal version of the syntax diagram.

```
►►──DROP ALERT CONDITIONS FOR QAPPLY SCHEMA schema──MONITOR QUALIFIER monitor-qualifier──────────────────►◄
```

Parameters

SCHEMA
Specifies the Q Apply schema that qualifies the process to be monitored. The default is ASN.

MONITOR QUALIFIER
Specifies the monitor qualifier grouping the alert conditions.

Appendix A. Frequently asked questions

This appendix contains answers to some of the issues that you might encounter while using the ASNCLP commands.

General ASNCLP issues

Question: What happens if I specify the **SET SERVER** command twice in my input file?

Response: The ASNCLP program overrides the server information specified by the first command with the server information specified by the second command.

Question: Why do I get an SQL0805N message when I connect to a remote z/OS system for the first time?

Response: You will get this message if you did not run the **db2 bind** command. See "Binding required when using ASNCLP with DB2 for z/OS" on page 16.

Question: Can I create a nickname if I am I using z/OS?

Answer: No, the ASNCLP program does not generate scripts to create nicknames on z/OS.

Question: How do I comment out ASNCLP commands in the script?

Answer: You can precede the command with a `#` sign or enclose a set of commands from `\*` to `*/`.

Question: How do I know if there were any API errors?

Answer: Check the ASNCLP log file for any errors or warnings.

SQL replication issues

Question: When I start ASNCLP, I get the following error:

```
Exception in thread "main" java.lang.NoClassDefFoundError:

com/ibm/db2/tools/repl/replapis/cmdline/Asnclp
```

Response: Ensure that the Java classpath is set correctly. For more information, see Chapter 2, "Using the ASNCLP program," on page 15.

Question: In the **CREATE CONTROL TABLES** clause, must I specify the **UW** keyword?

Response: The **UW** keyword needs to be specified only if you are dealing with a server on a workstation. After specifying the **UW** keyword, you can optionally specify a **UOW** table space name, following the **UOW** keyword.

Question: If I issue `set capture schema target xxx`, is it used in the **CREATE CONTROL TABLES** command?

Response: If SET schema TARGET is xxx, APPLY control tables will be created with schema xxx, not Capture.

Question: What is the requirement to create control tables on a federated system?

Response: You must issue the following two commands:

```
SET SERVER CAPTURE TO DB fedinfdb NONIBM SERVER rmtinfregres;
CREATE CONTROL TABLES FOR CAPTURE SERVER IN NONIBM SCHEMA "undjr14";
```

Note: When dealing with federated systems, you must specify the **NONIBM** server in the **SET SERVER** command before issuing any of the replication task commands for federated.

Question: Should I specify the key columns in the **CREATE MEMBER** command, if the target table already exists?

Response: Yes. You must specify the key columns using the **KEYS** keyword.

Question: Why does the ASNCLP program fail when I specify the **ALERTS** keyword in the **zos-ts-clause** of the **CREATE CONTROL TABLES** command?

Response: Check the **SET SERVER** command that you specified. Here are some basic rules to follow:

For Capture control server
Only UOW, PAGE and ROW can be specified.

For Apply control server
only PAGE and ROW can be specified.

For Monitor control server
Only ALERTS, PAGE and ROW can be specified.

Question: What happens to the parameters passed to the **CREATE REGISTRATION** command when the object is a view; that is, when I want to register a view?

Response: The parameters are ignored. The ASNCLP programs determines the necessary information from existing registration definitions.

Question: Is there a way to provide for a default table space naming prefix when generating a table space name?

Response: Currently, the only mechanism available is to use the **NAMING PREFIX** keyword and providing a prefix for the ASNCLP program to use.

Question: How does the table space **REUSE** keyword work?

Response: Here are some basic rules:

- If you do not specify the **IN** clause, then ASNCLP uses the DB2 defaults.
- If you do specify the **IN** clause is specified with a *tsname* value:
  - If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - If you want to reuse an existing table space, include the **REUSE** clause.
  - If you do not have a profile, specify the table space name with no profile, and the ASNCLP program assumes that the table space exists.

- If you specify a **NAMING PREFIX**, same logic as the **IN** clause, except that the command line generates the *tsname* value.
  - – If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile.
  - – If you want to reuse an existing table space, include the **REUSE** clause.
  - – If you do not have a profile, specify the table space name with no profile, and the ASNCLP program assumes that the table space exists.

Note: When working with non-IBM servers, you can specify an existing table space name or a segment name for remote sources that support those servers.

Question: What is the purpose of a **SET PROFILE** command?

Response: The profile allows you to set certain common DDL information that can be used by all replication task commands. For example, you can set the buffer pool name with the **BUFFERPOOL** keyword.

Question: How can I use the profile once it has been set?

Response: When you invoke the **SET PROFILE** command, you give it a name as follows:

```
SET PROFILE myprof options;
```

You can then use the profile in the **CREATE REGISTRATION** command as follows:

```
CREATE REGISTRATION (ibm.table1) DIFFERENTIAL REFRESH
   STAGE ibm.cd_table1 IN mytablespace
   CREATE USING PROFILE myprof;
```

Once the table space *mytablespace* is created, it will use the properties that you specified in the **SET PROFILE** command.

Question: What if I want to run my script before executing the next task command?

Response: Use the **SET RUN SCRIPT** command. For example:

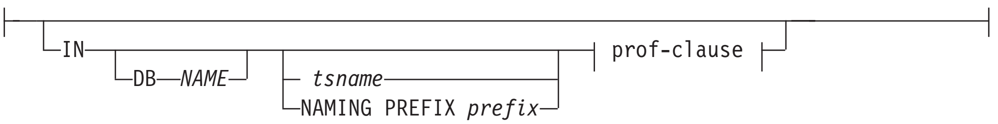
```
SET RUN SCRIPT NOW STOP ON SQL ERROR ON;
CREATE CONTROL TABLES options
CREATE REGISTRATION options
```

When the ASNCLP program reaches the **CREATE REGISTRATION** command, control tables are already created in the database.

Question: How does the **NAMING PREFIX** keyword work?

Response: Consider the following syntax from the **CREATE MEMBER** command:

trg-def-clause:



If the **NAMING PREFIX** keyword is specified, the prefix value provided in the *prefix* variable is used to generate the table space name. For example, if the target table name is *MYTGT* and the trg-def-clause is specified as follows:

```
IN NAMING PREFIX IBM CREATE USING PROFILE MYPROF
```

Then, the table space name will be *IBMMYTGT*.

Q replication issues

Question: Are there sample ASNCLP script files? If so, where can I find them?

Answer: The ASNCLP samples are in the SQLLIB\samples\repl\asnclp directory.

Question: Where can I find more information on the peer-to-peer samples?

Answer: For more detail, see the p2preadme.txt file also located in the SQLLIB\samples\repl\asnclp directory.

Question: I am trying to create Q Capture control tables but the script generated is for SQL Capture control tables. What do I need to do?

Answer: Only one type of replication definition commands (SQL replication or Q replication) can be issued from a single ASNCLP session. In this case, you would need to set the for a Q Replication session with the following command:

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

Question: When defining a profile, part of the command involve specifying table space options in the zos-tbs-clause is PERCENT OF SOURCE and PERCENT OF SOURCE ALLOC. How is the absolute value determine from the percentage?

Answer: PERCENT OF SOURCE is the percentage of the source table size, as indicated by the following columns:

- The column npages in SYSIBM.SYSTABLES if the source platform is OS/390
- The column npages in SYSSTAT.TABLES, if the source platform is LUOW

Note: This method will only work 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.

PERCENT OF SOURCE ALLOC is the percentage of the source table allocation; for example, the space that is allocated for the source table as indicated by the following information:

- The absolute size of the source table space provided by the table space containers if the source platform is LUOW and the source table space is a table space that is managed by database. If the source table table space is MANAGED BY SYSTEM, this method will not work.
- The maximum size of the source table space provided by the table space partitions if the source platform is OS/390 and the target platform is LUOW.
- The sum of all values of the primary quantity of the source table space partitions if the source and target platform is OS/390 and the unit is provided for the primary quantity of the target table space.
- The sum of all values of the secondary quantity of the source table space partitions if the source and target platform is OS/390 and the unit is provided for the secondary quantity of the target table space. Also, if the target platform is

LUOW and the container size is lower than the replication defined minimum of 2 MB, the container size is set to 2MB and an informational message is generated.

Question: If I were working under a Q replication session, what kinds of error would I see if I issue SQL replication commands? Would I be able to change to a SQL replication session?

Answer: You will either see an unexpected token or out of context error message.

Only one type of replication definition commands (SQL replication or Q replication) can be issued from a single ASNCLP session. In this case you would need to quit and start a new ASNCLP session.

Question: What the purpose of having bidirectional or peer-to-peer replication scripts?

Answer: The idea of having bidirectional or peer-to-peer replication scripts is to support creating multiple nodes in a subgroup. The scope of each node of a subgroup are enclosed to its own bidirectional or peer-to-peer replication script and they are loaded from the main ASNCLP script with the **LOAD MULTIDIR REPL SCRIPT** command.

Question: Is load type 1 for crossloader available in peer-to-peer replication?

Answer: No.

Question: If the name of the Q subscription for the keyword subname is an option, how does the ASNCLP program generate the default name?

Answer: The default name will be SRC TABLE NAME plus a four-digit counter.

Appendix B. Dotted decimal diagrams

In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, as they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number, for example 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. For example, if you hear the lines 3.1 USERID, 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with the dotted decimal number 3 is followed by a series of syntax elements with the dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Characters such as commas that are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. They might appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line might also show another symbol giving information about the syntax elements; all these symbols are explained below. For example, the lines 5.1* ,, 5.1 LASTRUN, 5.1 DELETE mean that if you use more than one of the syntax elements LASTRUN and DELETE, they must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % is the name of a syntax fragment, rather than a literal. For example, the line 2.1 %OP1 means that at this point, you should refer to the separate syntax fragment OP1. OP1, in the syntax from which this example was taken, gave a list of further options.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the escape character, which is a \ (backslash). For example, the * symbol can be used next to a dotted decimal number to mean that this syntax element can be repeated. If a syntax element actually starts with the * symbol, for example a syntax element * FILE with the dotted decimal number 3, it is given in the format 3 * FILE. If the format is 3* FILE, this means that there is a syntax element FILE, which can be repeated. If the format was 3* * FILE, this means that there is a syntax element * FILE, which can be repeated.

The words and symbols used next to the dotted decimal numbers are as follows:

- **?** means an optional syntax element. If a dotted decimal number is followed by the ? symbol, this means that all the syntax elements with that dotted decimal number, and any subordinate syntax elements that they each have, are optional. If there is only one syntax element with that dotted decimal number, the ? symbol appears on the same line as the syntax element, for example 5? NOTIFY.

If there is more than one syntax element with that dotted decimal number, the ? symbol appears on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional; you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

- ! means a default syntax element. If a dotted decimal number is followed by the ! symbol, appended to the last digit of the dotted decimal number, this means that this syntax element is the default of all the elements with the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a !. For example, if you hear the lines 2? FILE, 2.1! (KEEP), 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. If you include the FILE keyword but do not state your choice of option, the default option KEEP is applied. As well as the particular syntax element marked with the ! symbol, the default also applies to the next higher dotted decimal number. In the example above, the default applies to 2? FILE as well as to 2.1! (KEEP), meaning that if you omit the word FILE, the default FILE(KEEP) is used. However, you might instead hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), 2.1.1 (DELETE). As the default only applies to the next higher dotted decimal number, which in this case is 2.1, it does not apply to 2? FILE. In this case, if you omit the word FILE, nothing is used.
- * means a syntax element that is optional and can be repeated. If a dotted decimal number is followed by the * symbol, this means that this syntax element it is optional, and can be repeated. For example, if you hear the line 5.1* data-area, you know that you can include more than one data area, or you can include none. If you hear the lines 3*, 3 HOST, 3 STATE, you know that you can include HOST, STATE, both, or nothing. Note that if a dotted decimal number has an asterisk next to it, and there is only one item with that dotted decimal number, you can repeat that same item more than once. If a dotted decimal number has an asterisk next to it, and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the example above, you could write HOST STATE, but you could not write HOST HOST. The * symbol is equivalent to a loop back line in a railroad syntax diagram.
- + means a syntax element that must be included at least once, and can be repeated. If a dotted decimal number is followed by the + symbol, this means that this syntax element must be included at least once, and can be repeated. For example, if you hear the line 6.1+ data-area, you know that you must include at least one data area, and you can include more than one. If you hear the lines 2+, 2 HOST, 2 STATE, you know that you must include HOST, STATE, or both. As for the + symbol, you can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop back line in a railroad syntax diagram.

Dotted decimal format for syntax diagrams in this document

ALTER ADD COLUMN

```
1 ALTER ADD COLUMN USING SIGNAL(
2+ ,
2  colname
3  )
2 QSUB subname USING REPQMAPqmapname
2 XML PUB pubname
```

ALTER ALERT CONDITIONS FOR APPLY

```
1 ALTER ALERT CONDITIONS FOR APPLY  QUALIFIER qual-name
2? SET NAME set-name
3 MONITOR-QUALIFIER  mon-qual
2 NOTIFY
3 CONTACT  contact-name
3 GROUP  group-name

1 (
2 ADD
2 REMOVE
2 CHANGE
3 STATUS DOWN
3 ERRORS
3 WARNINGS
3 SUBSCRIPTIONS FAILING
3 SUBSCRIPTIONS DELAYED time
3 SUBSCRIPTIONS INACTIVE
3 SUBSCRIPTIONS REFRESHED
3 TRANSACTIONS REJECTED
3 REWORKED ROWS rows
3 LATENCY end-end-latency
3  )
```

ALTER ALERT CONDITIONS FOR CAPTURE

```
1 ALTER ALERT CONDITIONS FOR CAPTURE
2? SCHEMA cap-schema
3 MONITOR-QUALIFIER  mon-qual
2? NOTIFY
2.1 CONTACT  contact-name
2.1 GROUP  group-name
3 (
4+ ,
4 ADD
4 REMOVE
4 CHANGE
4.1 STATUS DOWN
4.1 STATUS LAST COMMIT time-secs
4.1 ERRORS
4.1 WARNINGS
4.1 CURRENT LATENCY latency
4.1 HISTORIC LATENCY latency
4.1 MEMORY memory
5  )
```

ALTER ALERT CONDITIONS FOR QAPPLY

```
1 ALTER ALERT CONDITIONS FOR\nQAPPLY
2? SCHEMA schema
2 MONITOR-QUALIFIER monitor-qualifier
3? notification-list-definition
4 (
5+ ,
5 ADD
5 REMOVE
5 CHANGE
5.1  STATUS DOWN
5.1 ERRORS
5.1 WARNINGS
5.1 LATENCY  seconds
5.1 EELATENCY  seconds
5.1 MEMORY  megabytes
5.1 EXCEPTIONS
5.1 SPILL QUEUES DEPTH  percentage
5.1 QUEUE DEPTH  queue-percent
5.1.1  )

notification-list-definition:

1 NOTIFY
2 CONTACT contact-name
2 GROUP group-name
```

ALTER ALERT CONDITIONS FOR QCAPTURE

```
1 ALTER ALERT CONDITIONS FOR\nQCAPTURE
2? SCHEMA schema
2 MONITOR-QUALIFIER monitor-qualifier
3? notification-list-definition
4 (
5+ ,
5 ADD
5 REMOVE
5 CHANGE
5.1  STATUS DOWN
5.1 ERRORS
5.1 WARNINGS
5.1 LATENCY  seconds
5.1 MEMORY  megabytes
5.1 TRANSACTION SIZE  megabytes
5.1 SUBSCRIPTIONS INACTIVE
5.1 QUEUE DEPTH  queue-percent
5.1.1  )

notification-list-definition:

1 NOTIFY
2 CONTACT contact-name
2 GROUP group-name
```

ALTER CONTACT

```
1 ALTER CONTACT
2 contact-name
3 EMAIL
3 PAGE
4  "email-address"
5? DESCRIPTION  "desc"
```

ALTER GROUP

```
1 ALTER GROUP
2 group-name
3? DESCRIPTION  "desc"
2 NEW CONTACTS
2.1+ ,
2.1  contact-name1
2 CONTACTS
2.1+ ,
2.1  contact-name2
2.1.1 ADD
2.1.1 REMOVE
```

ALTER MEMBER ADD COLS

```
1 ALTER MEMBER ADD COLS IN SETNAME  setname APPLYQUAL  applyq SOURCE
2? objowner .
3 objname
3 TARGET
4? objowner .
5 objname
```

```
4 COLS (  EXPRESSION "source-col-or-expr"
5? TARGET name
5.1?  +
5.1.1 )
```

ALTER PUBQMAP

```
1 ALTER PUBQMAP qmapname  USING options

options:

1? DESC "description"
2? MESSAGE CONTENT TYPE
2.1  T
2.1  R
3? SENDQ sendqname
4? ERROR ACTION
4.1  I
4.1  S
5? HEARTBEAT INTERVAL interval
6? MAX MESSAGE SIZE size
7? HEADER
7.1 NONE
7.1 MQ RFH2
```

ALTER QSUB (bidirectional Q replication)

```
1 ALTER QSUB SUBTYPE B
2? FROM NODE servername.schemaname
3? SOURCE source-clause
4? TARGET target-clause

1? FROM NODE servername.schemaname
2? SOURCE source-clause
3? TARGET target-clause

source-clause:

1? ALL CHANGED ROWS
1.1  N
1.1  Y
2? HAS LOAD PHASE
2.1  N
2.1  I
2.1  E

target-clause:

1? CONFLICT RULE
1.1  K
1.1  C
1.1  A
2? CONFLICT ACTION
2.1  I
2.1  F
2.1  D
2.1  S
2.1  Q
3? ERROR ACTION
3.1  Q
3.1  D
3.1  S
4? LOAD TYPE
4.1  0
4.1  2
4.1  3
5? OKSQLSTATES "sqlstates"
```

ALTER QSUB (peer-to-peer Q replication)

```
1 ALTER QSUB SUBTYPE P
2 SOURCE source-clause
3? TARGET target-clause

source-clause:

1? HAS LOAD PHASE
1.1  N
1.1  I
1.1  E

target-clause:

1? ERROR ACTION
1.1  Q
1.1  D
1.1  S
2? LOAD TYPE
2.1  0
2.1  2
2.1  3
3? OKSQLSTATES "sqlstates"
```

ALTER QSUB (unidirectional Q replication)

```
1 ALTER QSUB subname  REPLQMAP mapname
2? USING REPLQMAP mapname
3? DESC description
4?
4 USING OPTIONS  other-opt-clause

other-opt-clause:

1? SEARCH CONDITION "search_condition"
2? ALL CHANGED ROWS
2.1  N
2.1  Y
3? HAS LOAD PHASE
3.1  N
3.1  I
3.1  E
4? SUPPRESS DELETES
4.1  N
4.1  Y
5? CONFLICT ACTION
5.1  I
5.1  F
5.1  D
5.1  S
5.1  Q
6? ERROR ACTION
6.1  S
6.1  D
6.1  Q
7? OKSQLSTATES "sqlstates"
8? LOAD TYPE
8.1  0
8.1  1
8.1  2
8.1  3
```

ALTER REGISTRATION

```
1 ALTER REGISTRATION
2 ROW
2.1 % row-clause
2 ADD
2.1 %add-cols-clause

row-clause:

1 (
2+ ,
2? objowner.
2.1 objname
2.1.1 )
2? CONFLICT
2.1 NONE
2.1 STANDARD
2.1 ENHANCED
3?
3 UPDATE AS DELETE INSERT
3.1 OFF
3.1 ON
4? CAPTURE
4.1 ALL
4.1 CHANGES
5?
5 FORWARDING
5.1 OFF
5.1 ON
6? FULL REFRESH
6.1 ON
6.1 OFF
7? STOP ON ERROR
7.1 ON
7.1 OFF

add-cols-clause:

1? objowner.
2 objname
3? COLS (
3.1+ ,
3.1 colname
3.1.1? IMAGE
3.1.1.1 AFTER
3.1.1.1 BOTH
3.1.1.1 BEFORE
3.1.1.2 )
4? PREFIX
4.1 X
4.1 befimgpref
```

ALTER REPLQMAP

```
1 ALTER REPLQMAP qmapname  USING options

options:

1? DESC "description"
2? ADMINQ "admnqname"
3? RECVQ "recvqname"
4? SENDQ "sendqname"
5? NUM APPLY AGENTS num
6? MEMORY LIMIT limit
7? ERROR ACTION
```

```
7.1  I
7.1  S
8? HEARTBEAT INTERVAL interval
9? MAX MESSAGE SIZE size
```

ALTER SUBSCRIPTION SET

```
1 ALTER SUBSCRIPTION SET SETNAME  setname APPLYQUAL  applyq SETTYPE
2 R
2 U
2.1?
2.1 F ONLY
2.1 S ONLY
2 P
2? ACTIVATE
2.1 NO
2.1 YES
2.1 ONCE
2.1.1? TIMING
2.1.1.1 EVENT  name
2.1.1.1 INTERVAL mn
2.1.1.1 BOTH
2.1.1.1.1 EVENT name  INTERVAL mn
2.1.1.1 CONTINUOUS
3? BLOCKING minutes
4? COMMIT COUNT
4.1  n
4.1 NULL
```

ALTER XML PUB

```
1 ALTER XML PUB pubname  FOR
2? source_owner.
3  source_name
2? DESC "description"
3? PUBQMAP qmapname
4? OPTIONS opt-clause

opt-clause:

1? SEARCH CONDITION "search_cond"
2? ALL CHANGED ROWS
2.1  N
2.1  Y
3? BEFORE VALUES
3.1  N
3.1  Y
4? CHANGED COLS ONLY
4.1  Y
4.1  N
5? SUPPRESS DELETES
5.1  N
5.1  Y
6? TOPIC "topic"
```

CREATE ALERT CONDITIONS FOR APPLY

```
1 CREATE ALERT CONDITIONS FOR APPLY  QUALIFIER qual-name
2? SET NAME set-name
3 MONITOR-QUALIFIER  mon-qual
2 NOTIFY
3 CONTACT  contact-name
3 GROUP  group-name

1 (
2 STATUS DOWN
2 ERRORS
2 WARNINGS
2 SUBSCRIPTIONS FAILING
2 SUBSCRIPTIONS DELAYED time
2 SUBSCRIPTIONS INACTIVE
2 SUBSCRIPTIONS REFRESHED
2 TRANSACTIONS REJECTED
2 REWORKED ROWS rows
2 LATENCY  end-end-latency
3  )
```

CREATE ALERT CONDITIONS FOR CAPTURE

```
1 CREATE ALERT CONDITIONS FOR CAPTURE
2? SCHEMA cap-schema
3 MONITOR QUALIFIER  mon-qual
2 NOTIFY
3 CONTACT  contact-name
3 GROUP  group-name
3  (
4+ ,
4 STATUS DOWN
4 STATUS LAST COMMIT time-secs
4 ERRORS
4 WARNINGS
4 CURRENT LATENCY latency
4 HISTORIC LATENCY latency
4 MEMORY memory
4  )
```

CREATE ALERT CONDITIONS FOR QAPPLY

```
1 CREATE ALERT CONDITIONS FOR\nQAPPLY
2? SCHEMA schema
2 MONITOR-QUALIFIER monitor-qualifier
3? notification-list-definition
4 (
5+ ,
5  STATUS DOWN
5 ERRORS
5 WARNINGS
5 LATENCY  seconds
5 EELATENCY  seconds
5 MEMORY  megabytes
5 EXCEPTIONS
5 SPILL QUEUES DEPTH  percentage
5 QUEUE DEPTH  queue-percent
6  )

notification-list-definition:

1 NOTIFY
2 CONTACT contact-name
2 GROUP group-name
```

CREATE ALERT CONDITIONS FOR QCAPTURE

```
1 CREATE ALERT CONDITIONS FOR QCAPTURE
2? SCHEMA schema
3 MONITOR QUALIFIER monitor-qualifier  notification-list-definition  (
4+ ,
4 STATUS DOWN
4 ERRORS
4 WARNINGS
4 LATENCY  seconds
4 MEMORY  megabytes
4 TRANSACTION SIZE  megabytes
4 SUBSCRIPTIONS INACTIVE
4 QUEUE DEPTH  queue_percent
5  )

notification-list-definition:

1 NOTIFY
2 CONTACT contact-name
2 GROUP group-name
```

CREATE CONTACT

```
1+ ,
1 CREATE CONTACT
1.1 contact-name
1.2? GROUP  group-name
2 EMAIL
2 PAGE
3  "email-address"
4? DESCRIPTION  "desc"
```

CREATE CONTROL TABLES (Q replication)

```
1 CREATE CONTROL TABLES FOR
1.1 CAPTURE SERVER USING  capparms-clause
1.1 APPLY SERVER
1.1.1? USING  appparms-clause
2? IN
2.1 ZOS  zos-ts-clause
2.1 UW uw-ts-clause
3? ZOS
3.1  INDEX  zos-idx-clause

capparms-clause:

1 RESTARTQ  "rstqname"
2 ADMINQ  "admqname"
3? STARTMODE
3.1  WARMSI
3.1  COLD
3.1  WARMNS
3.1  WARMSA
4?
4 MEMORY LIMIT limit
5? AUTOSTOP
5.1  N
5.1  Y
6? MONITOR INTERVAL interval
7? MONITOR LIMIT monlimit
8? TRACE LIMIT trclimit
9? SIGNAL LIMIT siglimit
10? PRUNE INTERVAL prninterval
11? SLEEP INTERVAL sleepinterval
12? LOGREUSE
12.1  N
12.1  Y
13? LOGSTDOUT
13.1  N
13.1  Y
14? TERM
14.1  Y
14.1  N
15? CAPTURE PATH "capture_path"

apparms-clause:

1? MONITOR LIMIT monlimit
2? TRACE LIMIT trclimit
```

```
3? MONITOR INTERVAL interval
4? PRUNE INTERVAL prninterval
5? AUTOSTOP
5.1  N
5.1  Y
6? SLEEP INTERVAL sleepinterval
7? LOGREUSE
7.1  N
7.1  Y
8? LOGSTDOUT
8.1  N
8.1  Y
9? APPLY PATH "apply_path"
10? TERM
10.1  Y
10.1  N
11? PWDFILE "filename"
12? DEADLOCK RETRIES num
13? SIGNAL LIMIT siglimit

zos-ts-clause:

1? PAGE LOCK
1.1? DB  name
1.2  tsname
1.2 NAMING PREFIX prefix
1.3 %prof-clause
2? ROW LOCK
2.1? DB  name
2.2  tsname
2.2 NAMING PREFIX prefix
2.3 %prof-clause

uw-ts-clause:

1?
1 TBSPACE
1.1  tsname
1.1 NAMING PREFIX prefix
1.2 prof-clause

prof-clause:

1? CREATE
1.1?  USING PROFILE pname

zos-idx-clause:

1? CREATE  USING PROFILE pname
```

CREATE CONTROL TABLES (SQL replication)

```
1 CREATE CONTROL TABLES FOR
1.1 CAPTURE SERVER
1.1 APPLY CONTROL SERVER
1.1 MONITOR CONTROL SERVER
2? IN
2.1 ZOS  zos-ts-clause
2.1 UW uw-ts-clause
2.1 NONIBM fed-ts-clause

zos-ts-clause:

1?
1 UOW DB dbname
1.1  tsname
1.1 NAMING PREFIX prefix
```

```
1.2 %prof-clause
2?
2 ALERTS DB dbname
2.1  tsname
2.1 NAMING PREFIX prefix
2.2 prof-clause
3?
3 PAGE LOCK DB dbname
3.1  tsname
3.1 NAMING PREFIX prefix
3.2 prof-clause
4?
4 ROW LOCK DB dbname
4.1  tsname
4.1 NAMING PREFIX prefix
4.2 prof-clause

uw-ts-clause:

1?
1 UOW
1.1  tsname
1.1 NAMING PREFIX prefix
1.2 prof-clause
2?
2 OTHERS
2.1  tsname
2.1 NAMING PREFIX prefix
2.2 prof-clause

fed-ts-clause:

1?
1 OTHERS
1.1  tsname
1.1 NAMING PREFIX prefix
1.2 prof-clause
2 SCHEMA name

prof-clause:

1?
1 CREATE USING PROFILE pname
1 REUSE
```

CREATE GROUP

```
1 CREATE GROUP
2 group-name
3? DESCRIPTION  "desc"
2 CONTACTS
3+ ,
3  contact-name ,...
```

CREATE MEMBER

```
1 CREATE MEMBER IN SETNAME  setname APPLYQUAL  applyq
2? ACTIVATE
2.1 NO
2.1 YES
2.1 ONCE
2 SOURCE
3? objowner .
4 objname
5? %target-clause
6? TGT KEY CHANGE
6.1 OFF
```

```
6.1 ON
7? WHERE  "sql-where-stmts"
8? COLS
8.1!ALL REGISTERED
8.1?
8.1 INCLUDE (
8.1.1+ ,
8.1.1 EXPRESSION  "expr"
8.1.1.1? TARGET name
8.1.1.2 )
8.1 EXCLUDE(
8.1.1+ ,
8.1.1 colname
8.1.2 )
9?
9 KEYS (
9.1+ ,
9.1 keyname
9.1.1  +
9.1.1  -
9.1.2 )
10? %loadx-clause

target-clause:

1 TARGET
2?
2 NAME
2.1? owner .
2.2 name
2 NAMING PREFIX prefix
2? DEFINITION-| trg-def-clause |

trg-def-clause:

1? IN
1.1? DB NAME
1.2  tsname
1.2 NAMING PREFIX prefix
1.3 %prof-clause
2?
2 TYPE
2.1 PIT
2.1 USERCOPY
2.1 BASEAGGREGATE
2.1 CHANGEAGGREGATE
2.1 REPLICA
2.1.1 %replica-clause
2.1 CCD
2.1.1 %ccd-clause
2.1.2 EXTERNAL
2.1.2 INTERNAL

prof-clause:

1?
1 CREATE USING PROFILE pname
1 REUSE

replica-clause:

1? CD
1.1? cdowner .
1.2 cdname
2? IN
2.1? DB NAME
2.2  tsname
```

```
2.2 NAMING PREFIX prefix
2.3 %prof-clause
3?
3 UPDATE AS DELETE INSERT
3.1 OFF
3.1 ON
4? FORWARDING
4.1 OFF
4.1 ON
5? FULL REFRESH
5.1 ON
5.1 OFF
6? STOP ON ERROR
6.1 ON
6.1 OFF
```

```
ccd-clause:

1 AS SOURCE
1 WITH UOW COLS
1.1 ALL
1.1 cols-clause
1 COMPLETE
1.1 ON
1.1 OFF
2 CONDENSED
3 ON
3 OFF
```

```
cols-clause:

1 (
2+ ,
2 colname
3 )
```

```
loadx-clause:

1?
1 LOADX TYPE
1.1 NO ASNLOAD
1.1 USER DEFINED
1.1 CROSSLOADER LOAD SRC NICKNAME owner.tablename
1.1 LOAD EXPORT
1.1 IMPORT EXPORT
```

CREATE REGISTRATION

```
1 CREATE REGISTRATION (
2+ ,
2? objowner.
2.1 objname
2.2? RMTJRN LIB lib NAME name
2.2.1 )
2 DIFFERENTIAL REFRESH
2.1? %diff-ref-clause
2 FULL REFRESH ONLY
```

```
diff-ref-clause:

1? STAGE
1.1? cd_or_ccd_owner .
1.2 cd_or_ccd_name
2? CONDENSED
2.1 ON
2.1 OFF
3? NONIBM
```

```
3.1 %fed-clause
4? IN
4.1? DB NAME
4.2  tsname
4.2 NAMING PREFIX prefix
4.3 %prof-clause
5? COLS
5.1 %capcol-clause
6? OPTIONS
6.1 %opt-clause

fed-clause:

1?  remoteccdowner.
2 remoteccdname

prof-clause:

1?  CREATE USING PROFILE pname

capcol-clause:

1 ALL IMAGE
2 AFTER
2 BOTH
2? (
2.1+ ,
2.1 colname
2.1.1? IMAGE
2.1.1.1 AFTER
2.1.1.1 BOTH
2.1.1.2 )
3? PREFIX
3.1 X
3.1 befimgpref

opt-clause:

1? CONFLICT
1.1 NONE
1.1 STANDARD
1.1 ENHANCED
2?
2 UPDATE AS DELETE INSERT
2.1 OFF
2.1 ON
3? CAPTURE
3.1 ALL
3.1 CHANGES
4?
4 FORWARDING
4.1 OFF
4.1 ON
5? FULL REFRESH
5.1 ON
5.1 OFF
6? STOP ON ERROR
6.1 ON
6.1 OFF
```

CREATE PUBQMAP

```
1 CREATE PUBQMAP qmapname
2? DESC "description"
3 USING SENDQ "sendqname"
4? MESSAGE CONTENT TYPE
4.1  T
```

```
4.1  R
5? ERROR ACTION
5.1  I
5.1  S
6? HEARTBEAT INTERVAL interval
7? MAX MESSAGE SIZE size
8? HEADER
8.1 NONE
8.1 MQ RFH2
```

CREATE QSUB (bidirectional Q replication)

```
1 CREATE QSUB SUBTYPE B
2? FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause
2? FROM NODE servername.schemaname SOURCE source-clause TARGET target-clause

source-clause:

1? ALL CHANGED ROWS
1.1  N
1.1  Y
2? HAS LOAD PHASE
2.1  N
2.1  I
2.1  E

target-clause:

1? CONFLICT RULE
1.1  K
1.1  C
1.1  A
2? CONFLICT ACTION
2.1  I
2.1  F
2.1  D
2.1  S
2.1  Q
3? ERROR ACTION
3.1  Q
3.1  D
3.1  S
4? LOAD TYPE
4.1  0
4.1  2
4.1  3
5? OKSQLSTATES "sqlstates"
```

CREATE QSUB (peer-to-peer Q replication)

```
1 CREATE QSUB SUBTYPE P
2? SOURCE source-clause
3? TARGET target-clause

source-clause:

1? HAS LOAD PHASE
1.1  N
1.1  I
1.1  E

target-clause:

1? ERROR ACTION
1.1  Q
1.1  D
1.1  S
```

```
2? LOAD TYPE
2.1  0
2.1  2
2.1  3
3? OKSQLSTATES "sqlstates"
```

CREATE QSUB (unidirectional Q replication)

```
1 CREATE QSUB
2? SUBTYPE  U
3? USING REPLQMAP  name
2+ ,
2 (
2.1? SUBNAME  subname
2.1.1? DESC  "desc"
2.1.2? REPLQMAP  name
2.1.3 src-clause
2.1.4? trg-clause
2.1.5 )


src-clause:

1? source_owner.
1.1 source_name
1? SRC OWNER LIKE  "predicate1"
1.1 SRC NAME LIKE  "predicate2"
1 SRC ALL
2? OPTIONS opt-clause

opt-clause:

1? SEARCH CONDITION "search_condition"
2? ALL CHANGED ROWS
2.1  N
2.1  Y
3? HAS LOAD PHASE
3.1  N
3.1  I
3.1  E
4? SUPPRESS DELETES
4.1  N
4.1  Y

trg-clause:

1? EXIST
2?
2 TARGET
2.1?
2.1 NAME
2.1.1? owner .
2.1.2 target_name
2.1 NAMING PREFIX prefix
3? IN
3.1? DB name
3.1.1  tsname
3.1.1 NAMING PREFIX prefix
3.1.2? prof-clause
4? TYPE
4.1 USERTABLE
4.1 STOREDPROC
5? TRGCOLS
5.1 ALL
5.1 INCLUDE (
5.1.1+ ,
5.1.1 trgcolname
5.1.1.1? srccolname
5.1.1.2  )
5.1 EXCLUDE (
5.1.1+ ,
5.1.1  trgcolname
5.1.2  )
6? KEYS
6.1?
6.1  indexowner.indexname
6.1 NAMING PREFIX prefix
6.2  (
6.3+ ,
6.3  keyname
6.3.1?
6.3.1  +
6.3.1  -
6.3.2  )
7? ZOS INDEX CREATE USING PROFILE pname
8?
8 NICKNAME
8.1  owner.nickname
8.1 NAMING PREFIX prefix
8 NEW NICKNAME RMT SERVERNAME  srvname
8.1  owner.nickname
8.1 NAMING PREFIX prefix
9? CONFLICT ACTION
9.1  I
9.1  F
9.1  D
9.1  S
9.1  Q
10? ERROR ACTION
10.1  S
10.1  D
10.1  Q
11? OKSQLSTATES "sqlstates"
12? LOAD TYPE
12.1  0
12.1  1
12.1  2
12.1  3

prof-clause:

1? CREATE
1.1? USING PROFILE pname
```

CREATE REPLQMAP

```
1 CREATE REPLQMAP qmapname
2? DESC "description"
3 USING ADMINQ "admnqname"  RECVQ "recvqname"  SENDQ "sendqname"
4? NUM APPLY AGENTS num
5? MEMORY LIMIT limit
6? ERROR ACTION
6.1 I
6.1 S
7? HEARTBEAT INTERVAL interval
8? MAX MESSAGE SIZE size
```

CREATE STMT

```
1 CREATE STMT IN SETNAME  setname APPLYQUAL  applyq
2? SETTYPE
2.1 R
2.1 U
2.1 P
```

```
3! SQL "statement"
3?
3 PROC  "procname"
4? NUMBER  stmtnumber
5 EXECUTE
6 AT SOURCE
6 AFTER AT TARGET
6 BEFORE AT TARGET
6? SQLSTATES "states"
```

CREATE SUBSCRIPTION SET

```
1 CREATE SUBSCRIPTION SET SETNAME  setname APPLYQUAL  applyq
2? ACTIVATE
2.1 NO
2.1 YES
2.1 ONCE
2? SETTYPE
2.1 R
2.1 U
2.1 P
3? TIMING
3.1 EVENT  name
3.1 INTERVAL mn
3.1 BOTH
3.1.1 EVENT name  INTERVAL mn
3.1 CONTINUOUS
4? START DATE "yyyy-mm-dd" TIME "hh:mm:ss.ffffff"
5? NONIBM SOURCE SERVER srvrname
6? BLOCKING minutes
7? COMMIT COUNT n
```

CREATE XML PUB

```
1 CREATE XML PUB
2? USING PUBQMAP qmapname
2? 2+ ,
(
2.1? PUBNAME pubname
2.2? DESC "desc"
2.3? PUBQMAP name
2.4 src-clause
2.5? opt-clause
2.6 )

src-clause:

1? source_owner.
1.1 source_name
1? SRC OWNER LIKE  "predicate1"
1.1? SRC NAME LIKE  "predicate2"
1 SRC ALL
2? COLS col-cause

col-cause:

1 ALL
1 INCLUDE (
1.1+ ,
1.1  COLNAME
1.2  )
1 EXCLUDE (
1.1+ ,
1.1  COLNAME
1.2  )
2?
2 IS_KEY (
2.1+ ,
2.1  COLNAME
2.2  )

opt-clause:

1? SEARCH CONDITION "search_cond"
2? ALL CHANGED ROWS
2.1  N
2.1  Y
3? BEFORE VALUES
3.1  N
3.1  Y
4? CHANGED COLS ONLY
4.1  Y
4.1  N
5? HAS LOAD PHASE
5.1  N
5.1  E
6? SUPPRESS DELETES
6.1  N
6.1  Y
7? TOPIC "topic"
```

DELEGATE CONTACT

```
1 DELEGATE CONTACT
2 contact-name1
3  TO  contact-name2 FROM "start-date"
2 TO "end-date"
```

DROP ALERT CONDITIONS FOR APPLY

```
1 DROP ALERT CONDITIONS FOR APPLY QUALIFIER  apply-qual
1 MONTITOR QUALIFIER  mon-qual
```

DROP ALERT CONDITIONS FOR CAPTURE

```
1 DROP ALERT CONDITIONS FOR CAPTURE  SCHEMA cap-schema
1 MONTITOR QUALIFIER mon-qual
```

DROP ALERT CONDITIONS FOR QAPPLY

```
1 DROP ALERT CONDITIONS FOR QAPPLY SCHEMA schema
1 MONITOR QUALIFIER monitor-qualifier
```

DROP ALERT CONDITIONS FOR QCAPTURE

```
1 DROP ALERT CONDITIONS FOR QCAPTURE SCHEMA schema
1 MONITOR QUALIFIER monitor-qualifier
```

DROP CONTACT

```
1 DROP CONTACT
2 contact-name1
3? SUBSTITUTE WITH  contact-name2
```

DROP CONTROL TABLES (Q replication)

```
1 DROP CONTROL TABLES ON
1.1 CAPTURE SERVER
1.1 APPLY SERVER
```

DROP CONTROL TABLES (SQL replication)

```
1 DROP CONTROL TABLES ON
1.1 CAPTURE SERVER
1.1 APPLY CONTROL SERVER
1.1 MONITOR CONTROL SERVER
2? ARCHLEVEL
2.1 0801
2.1 0201
2.1 0805
3? NONIBM SCHEMA name
```

DROP GROUP

```
1 DROP GROUP
2 group-name
```

DROP MEMBER

```
1 DROP MEMBER FROM SETNAME  setname APPLYQUAL  applyq SOURCE
2? objowner .
3 objname
3 TARGET
4? objowner .
5 objname
```

DROP PUBQMAP

```
1 DROP PUBQMAP qmapname
```

DROP QSUB

```
1 DROP
2? ALL
3 QSUB
4? USING REPLQMAP qmapnamex
2?
2+ ,
2 (SUBNAME subname
2.1? USING REPLQMAP qmapname
2.2 )
2 FOR SUBNAME LIKE "predicate"
```

DROP REGISTRATION

```
1 DROP REGISTRATION (
2+ ,
2? objowner
2.1 objname
2.1.1 )
```

DROP REPLQMAP

```
1 DROP REPLQMAP qmapname
```

DROP STMT

```
1 DROP STMT FROM SETNAME  setname APPLYQUAL  applyq
2? SETTYPE
2.1 R
2.1 U
2.1 P
3 NUMBER (
4+ ,
4  number
5 )
```

DROP SUBGROUP

```
1 DROP SUBGROUP
```

DROP SUBSCRIPTION SET

```
1 DROP SUBSCRIPTION SET SETNAME  setname APPLYQUAL  applyq
```

DROP SUBTYPE (bidirectional Q replication)

```
1 DROP SUBTYPE
2  B
2 QSUBS
```

DROP SUBTYPE (peer-to-peer Q replication)

```
1 DROP SUBTYPE  P
2 QSUBS
```

DROP XML PUB

```
1 DROP XML PUB
2? ALL
2?
2+ ,
2 (PUBNAME  pubname)
2 FOR PUBNAME LIKE "predicate"
```

LOAD DONE

```
1 LOAD DONE  QSUB  SUBNAME subname
```

LOAD MULTIDIR REPL SCRIPT

```
1 LOAD MULTIDIR REPL SCRIPT "filelocation/filename"
```

OFFLINE LOAD

```
1 OFFLINE LOAD
2 BEFORE
2 AFTER
2 SETNAME setname APPLYQUAL  applyq
```

PROMOTE REGISTRATION

```
1 PROMOTE REGISTRATION (
2+ ,
2? objowner .
2.1 objname
2.1.1 )
2.1.1.1? USING
2.1.1.1.1 % new-clause

new-clause:

1? SOURCE DB alias
2? CAPTURE SCHEMA name
2 TABLE
2.1 %tbl-clause
2 VIEW
2.1 % view-clause

tbl-clause:

1? CD SCHEMA name
2? CREATE SOURCE WITH SCHEMA name
```

```
view-clause:

1? CD SCHEMA FOR
1.1? VIEW name
1.2? SOURCE TABLE name
2? CREATE SOURCE VIEW
2.1? WITH UNREGISTERED BASE TABLES
2.2? USING SCHEMA name
```

PROMOTE SUBSCRIPTION SET

```
1 PROMOTE SUBSCRIPTION SET  SETNAME setname APPLYQUAL  applyq
2? USING new-clause

new-clause:

1? CAPTURE SCHEMA FOR
1.1? SOURCE name
1.2?  REPLICA name
2?
2 DB FOR
2.1? SOURCE alias
2.2?  TARGET  alias
2.3?  CONTROL  alias
3? APPLYQUAL name
4? SETNAME name
5? SOURCE SCHEMA name
6?  TARGET
6.1? SCHEMA name
6.2? CD SCHEMA name
```

SET APPLY SCHEMA

```
1 SET APPLY SCHEMA
2 TO DEFAULT
2 applyschema
```

SET CAPTURE SCHEMA (Q replication)

```
1 SET CAPTURE SCHEMA SOURCE
1.1 TO
1.1.1 DEFAULT
1.1.1 NULLS
1.1 capschema
```

SET CAPTURE SCHEMA (SQL replication)

```
1 SET CAPTURE SCHEMA
1.1 SOURCE
1.1 TARGET
1.2 TO
1.2.1 DEFAULT
1.2.1 NULLS
1.2 capschema
```

SET CONNECTION

```
1 SET CONNECTION
2? SUBNAME subscription-name
2 SOURCE servername.schemaname TARGET servername.schemaname
3 REPLQMAP queue-map-name
```

SET DROP (Q replication)

```
1 SET DROP TARGET
1.1 NEVER
1.1 ALWAYS

1 SET DROP
1.1 TARGET
1.1 CONTROL TABLES
2 TABLESPACE
3 WHEN EMPTY
3 NEVER
```

SET DROP (SQL replication)

```
1 SET DROP TARGET
1.1 ALWAYS
1.1 NEVER

1 SET DROP
1.1 CD
1.1 CCD
1.1 TARGET
1.1 CONTROL TABLES
2 TABLESPACE
3 WHEN EMPTY
3 NEVER
```

SET LOG

```
1 SET LOG "logfilename"
```

SET MULTIDIR SCHEMA

```
1 SET MULTIDIR SCHEMA TO
2 DEFAULT
2 servername.schemaname
```

SET OUTPUT (Q replication)

```
1 SET OUTPUT
2? CAPTURE SCRIPT "capfname"
3? TARGET SCRIPT "trgfname"
4? MONITOR SCRIPT "monfname"
5? MULTIDIR
```

SET OUTPUT (SQL replication)

```
1 SET OUTPUT
2? CAPTURE SCRIPT "capfname"
3? CONTROL SCRIPT "cntlfname"
4? TARGET SCRIPT "trgfname"
5? MONITOR SCRIPT "monfname"
```

SET PROFILE (Q replication)

```
1 SET PROFILE name
2 prof-clause
2 UNDO

prof-clause:

1 FOR OBJECT
2 TARGET
2 QCNTL TBLS
2 PAGE LOCK
2 ROW LOCK
```

```
3 TABLESPACE OPTIONS
3.1 %zos-tbs-clause
3.1 %uw-tbs-clause
3 INDEX OPTIONS
3.1 %zos-idx-clause

zos-tbs-clause:

1 ZOS
2? DB name
3? BUFFERPOOL name
4? ENCODING
4.1 EBCDIC
4.1 ASCII
4.1 UNICODE
2? STOGROUP name
2.1? PRIQTY
2.1.1 ABSOLUTE n
2.1.1 PERCENT OF SOURCE n
2.1.1 PERCENT OF SOURCE ALLOC n
2.1.2? SECQTY
2.1.2.1 ABSOLUTE m
2.1.2.1 PERCENT OF SOURCE m
2.1.2.1 PERCENT OF SOURCE ALLOC m

uw-tbs-clause:

1 UW
2? BUFFERPOOL name
3? PAGESIZE n
4? USING
4.1 FILE
4.1 DEVICE
4.1.1 "container"
4.1.2 SIZE n
4.1.2.1 PAGES
4.1.2.1 KILO
4.1.2.1 MEGA
4.1.2.1 GIGA
4.1.2 PERCENT OF SOURCE n
4.1.2 PERCENT OF SOURCE ALLOC m

zos-idx-clause:

1 ZOS
2? BUFFERPOOL name
2? STOGROUP name
2.1? PRIQTY
2.1.1 ABSOLUTE n
2.1.1 PERCENT OF SOURCE n
2.1.1 PERCENT OF SOURCE ALLOC n
2.1.2? SECQTY
2.1.2.1 ABSOLUTE m
2.1.2.1 PERCENT OF SOURCE m
2.1.2.1 PERCENT OF SOURCE ALLOC m
```

SET PROFILE (SQL replication)

```
1 SET PROFILE name
2 prof-clause
2 UNDO

prof-clause:

1 FOR OBJECT
2 CD
```

```
2 CCD
2 TARGET
2 UOW
2 OTHERS
2 PAGE LOCK
2 ROW LOCK
3 TABLESPACE OPTIONS
4 %zos-tbs-clause
4 %uw-tbs-clause

zos-tbs-clause:

1 ZOS
2? DB name
3? BUFFERPOOL name
4? ENCODING
4.1 EBCDIC
4.1 ASCII
4.1 UNICODE
2? STOGROUP name
2.1? PRIQTY
2.1.1  ABSOLUTE
2.1.1 PERCENT OF SOURCE
2.1.2  n
2.1.3? SECQTY
2.1.3.1  ABSOLUTE
2.1.3.1 PERCENT OF SOURCE
2.1.3.2  m

uw-tbs-clause:

1 UW
2? BUFFERPOOL name
3? PAGESIZE n
4? USING
4.1 FILE
4.1 DEVICE
4.1.1 "container"
4.1.2 SIZE n
4.1.2.1 PAGES
4.1.2.1 KILO
4.1.2.1 MEGA
4.1.2.1 GIGA
4.1.2 PERCENT OF SOURCE n
```

SET QMANAGER

```
1 SET QMANAGER   "mgrname"  FOR
2 CAPTURE SCHEMA
2 APPLY SCHEMA
2 MULTIDIR servername.schemaname
```

SET REFERENCE TABLE

```
1 SET REFERENCE TABLE USING SCHEMA servername.schemaname
1 USES TABLE tableowner.tablename
```

SET RUN SCRIPT

```
1 SET RUN SCRIPT
1.1 LATER
1.1 NOW STOP ON SQL ERROR
1.1.1 ON
1.1.1 OFF
```

SET SERVER (bidirectional and peer-to-peer Q replication)

```
1 SET SERVER
2 CAPTURE
2 TARGET
2 MULTIDIR
2 TO
3 NULLS
3 DB dbalias
3 DBALIAS aliasname
3.1?
3.1 DBNAME dbname
3.2 other-options

other-options:

1? ID userid
2? PASSWORD pwd
```

SET SERVER (event publishing)

```
1 SET SERVER CAPTURE  TO
2 NULLS
2 DB dbalias
2 DBALIAS aliasname
2.1?
2.1 DBNAME dbname
2.2 other-options

other-options:

1? ID userid
2? PASSWORD pwd
```

SET SERVER (Replication Alert Monitor)

```
1 SET SERVER MONITOR  TO
2 NULLS
2 DB dbalias
2 DBALIAS aliasname
2.1?
2.1 DBNAME dbname
2.2 other-options

other-options:

1? ID userid
2? PASSWORD pwd
```

SET SERVER (SQL replication)

```
1 SET SERVER
2 ALL
2 REMOTE SOURCE
2 CAPTURE
2 CONTROL
2 TARGET
3  TO
2 NULLS
2 DB dbalias
2 DBALIAS aliasname
2.1?
2.1 DBNAME dbname
2.1 NONIBM SERVER remsrvr
2.2 other-options

other-options:
```

```
1? AS400 HOSTNAME "hostname"
2? ID userid
3? PASSWORD pwd
```

SET SERVER (unidirectional Q replication)

```
1 SET SERVER
2 CAPTURE
2 TARGET
2 TO
3 NULLS
3 DB dbalias
3 DBALIAS aliasname
3.1?
3.1 DBNAME dbname
3.2 other-options

other-options:

1? ID userid
2? PASSWORD pwd
```

SET SESSION

```
1 ASNCLP SESSION SET TO
2 SQL REPLICATION
2 Q REPLICATION
```

SET SUBGROUP

```
1 SET SUBGROUP subgroup-name
```

SET TABLES

```
1 SET TABLES(
2+ ,
2  servername.schemaname.tableowner.tablename
3  )
```

SET TRACE

```
1 SET TRACE
1.1 OFF
1.1 ON
```

SHOW SET ENV

```
1 SHOW SET ENV
```

START QSUB

```
1 START QSUB
2 SUBNAME subname
2 FOR SUBNAME LIKE "%text%"
```

START XML PUB

```
1 START XML PUB
2 PUBNAME pubname
2 FOR PUBNAME LIKE "%text%"
```

STOP QSUB

```
1 STOP QSUB
2 SUBNAME subname
2 FOR SUBNAME LIKE "%text%"
```

STOP XML PUB

```
1 STOP XML PUB
2 PUBNAME pubname
2 FOR PUBNAME LIKE "%text%"
```

SUBSTITUTE CONTACT

```
1 SUBSTITUTE CONTACT
2 contact-name1
3  WITH  contact-name2
```

DB2 Information Integrator documentation

This topic provides information about the documentation that is available for DB2 Information Integrator. The tables in this topic provide the official document title, form number, and location of each PDF book. To order a printed book, you must know either the official book title or the document form number. Titles, file names, and the locations of the DB2 Information Integrator release notes and installation requirements are also provided in this topic.

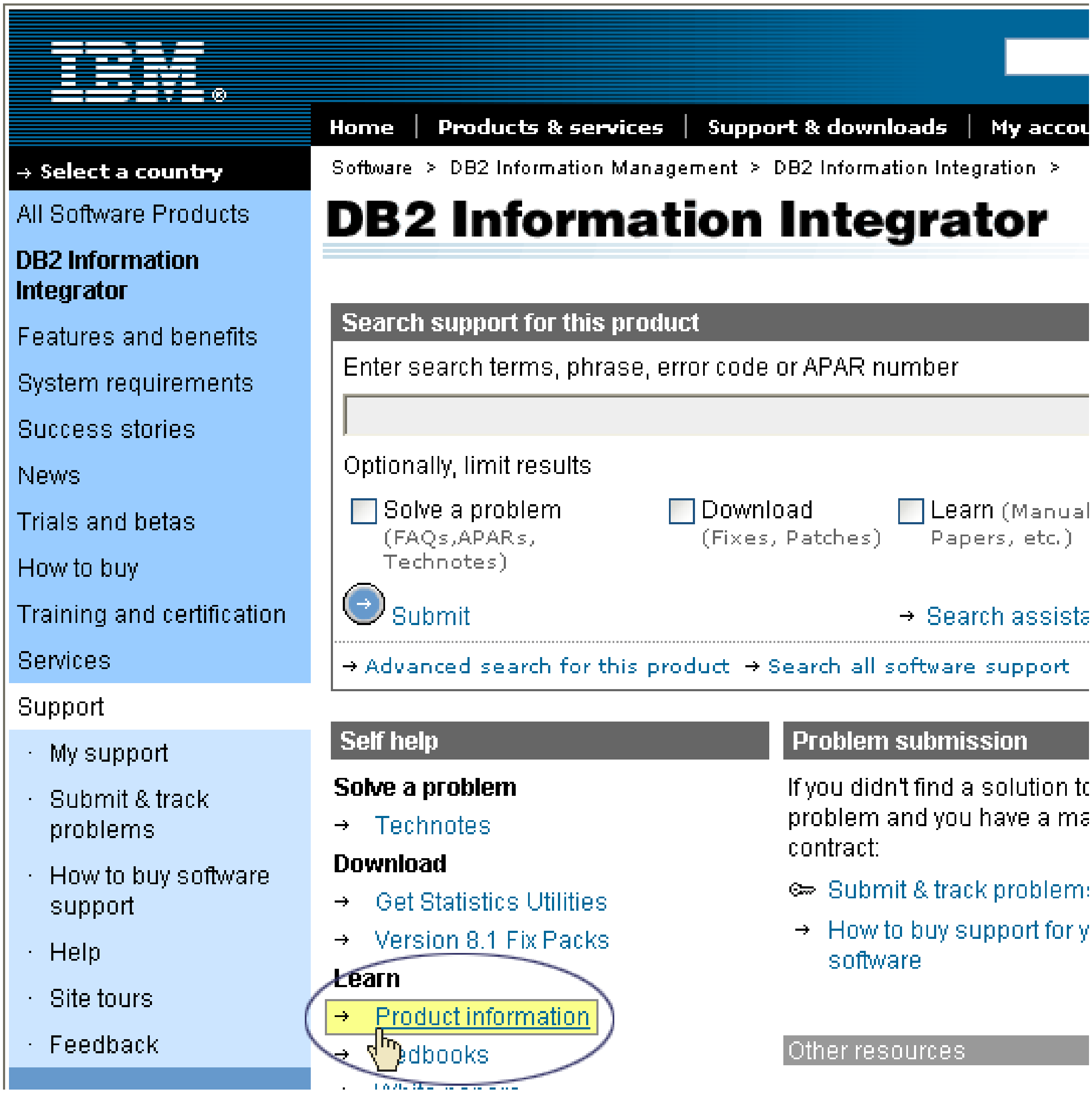
This topic contains the following sections:

- Accessing DB2 Information Integrator documentation
- Documentation for replication function on z/OS
- Documentation for event publishing function for DB2 Universal Database on z/OS
- Documentation for event publishing function for IMS and VSAM on z/OS
- Documentation for event publishing and replication function on Linux, UNIX, and Windows
- Documentation for federated function on z/OS
- Documentation for federated function on Linux, UNIX, and Windows
- Documentation for enterprise search on Linux, UNIX, and Windows
- Release notes and installation requirements

Accessing DB2 Information Integrator documentation

All DB2 Information Integrator books and release notes are available in PDF files from the DB2 Information Integrator Support Web site at www.ibm.com/software/data/integration/db2ii/support.html.

To access the latest DB2 Information Integrator product documentation, from the DB2 Information Integrator Support Web site, click on the Product Information link, as shown in Figure 1 on page 204.


IBM
Home | Products & services | Support & downloads | My accou
→ Select a country
Software > DB2 Information Management > DB2 Information Integration >
All Software Products
DB2 Information Integrator
Features and benefits
System requirements
Success stories
News
Trials and betas
How to buy
Training and certification
Services
Support
· My support
· Submit & track problems
· How to buy software support
· Help
· Site tours
· Feedback
DB2 Information Integrator
Search support for this product
Submit
→ Search assista
→ Advanced search for this product
→ Search all software support
Self help
Solve a problem
→ Technotes
Download
→ Get Statistics Utilities
→ Version 8.1 Fix Packs
Learn
→ Product information
→ Redbooks
Problem submission
Submit & track problem:
→ How to buy support for y software
Other resources

Enter search terms, phrase, error code or APAR number

Optionally, limit results

□ Solve a problem (FAQs,APARs, Technotes) □ Download (Fixes, Patches) □ Learn (Manual Papers, etc.)

If you didn't find a solution tc problem and you have a ma contract:

Figure 1. Accessing the Product Information link from DB2 Information Integrator Support Web site

You can access the latest DB2 Information Integrator documentation, in all supported languages, from the Product Information link:

- DB2 Information Integrator product documentation in PDF files
- Fix pack product documentation, including release notes
- Instructions for downloading and installing the DB2 Information Center for Linux, UNIX, and Windows
- Links to the DB2 Information Center online

Scroll though the list to find the product documentation for the version of DB2 Information Integrator that you are using.

The DB2 Information Integrator Support Web site also provides support documentation, IBM Redbooks, white papers, product downloads, links to user groups, and news about DB2 Information Integrator.

You can also view and print the DB2 Information Integrator PDF books from the *DB2 PDF Documentation* CD.

To view or print the PDF documentation:

1. From the root directory of the *DB2 PDF Documentation* CD, open the index.htm file.
2. Click the language that you want to use.
3. Click the link for the document that you want to view.

Documentation about replication function on z/OS

Table 10. DB2 Information Integrator documentation about replication function on z/OS

Name	**Form number**	**Location**
ASNCLP Program Reference for Replication and Event Publishing	N/A	DB2 Information Integrator Support Web site
Introduction to Replication and Event Publishing	GC18-7567	DB2 Information Integrator Support Web site
Migrating to SQL Replication	N/A	DB2 Information Integrator Support Web site
Replication and Event Publishing Guide and Reference	SC18-7568	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Replication Installation and Customization Guide for z/OS	SC18-9127	DB2 Information Integrator Support Web site
SQL Replication Guide and Reference	SC27-1121	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Tuning for Replication and Event Publishing Performance	N/A	DB2 Information Integrator Support Web site
Tuning for SQL Replication Performance	N/A	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Standard Edition, Advanced Edition, and Replication for z/OS	N/A	• In the DB2 Information Center, **Product Overviews** > **Information Integration** > **DB2 Information Integrator overview** > **Problems, workarounds, and documentation updates** • DB2 Information Integrator Installation launchpad • DB2 Information Integrator Support Web site • The *DB2 Information Integrator* product CD

Documentation about event publishing function for DB2 Universal Database on z/OS

Table 11. DB2 Information Integrator documentation about event publishing function for DB2 Universal Database on z/OS

Name	Form number	Location
ASNCLP Program Reference for Replication and Event Publishing	N/A	DB2 Information Integrator Support Web site
Introduction to Replication and Event Publishing	GC18-7567	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Replication and Event Publishing Guide and Reference	SC18-7568	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Tuning for Replication and Event Publishing Performance	N/A	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Standard Edition, Advanced Edition, and Replication for z/OS	N/A	• In the DB2 Information Center, **Product Overviews** > **Information Integration** > **DB2 Information Integrator overview** > **Problems, workarounds, and documentation updates** • DB2 Information Integrator Installation launchpad • DB2 Information Integrator Support Web site • The *DB2 Information Integrator* product CD

Documentation about event publishing function for IMS and VSAM on z/OS

Table 12. DB2 Information Integrator documentation about event publishing function for IMS and VSAM on z/OS

Name	Form number	Location
Client Guide for Classic Federation and Event Publisher for z/OS	SC18-9160	DB2 Information Integrator Support Web site
Data Mapper Guide for Classic Federation and Event Publisher for z/OS	SC18-9163	DB2 Information Integrator Support Web site
Getting Started with Event Publisher for z/OS	GC18-9186	DB2 Information Integrator Support Web site
Installation Guide for Classic Federation and Event Publisher for z/OS	GC18-9301	DB2 Information Integrator Support Web site
Operations Guide for Event Publisher for z/OS	SC18-9157	DB2 Information Integrator Support Web site

Table 12. DB2 Information Integrator documentation about event publishing function for IMS and VSAM on z/OS (continued)

Name	Form number	Location
Planning Guide for Event Publisher for z/OS	SC18-9158	DB2 Information Integrator Support Web site
Reference for Classic Federation and Event Publisher for z/OS	SC18-9156	DB2 Information Integrator Support Web site
System Messages for Classic Federation and Event Publisher for z/OS	SC18-9162	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Event Publisher for IMS for z/OS	N/A	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Event Publisher for VSAM for z/OS	N/A	DB2 Information Integrator Support Web site

Documentation about event publishing and replication function on Linux, UNIX, and Windows

Table 13. DB2 Information Integrator documentation about event publishing and replication function on Linux, UNIX, and Windows

Name	Form number	Location
ASNCLP Program Reference for Replication and Event Publishing	N/A	DB2 Information Integrator Support Web site
Installation Guide for Linux, UNIX, and Windows	GC18-7036	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Introduction to Replication and Event Publishing	GC18-7567	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Migrating to SQL Replication	N/A	DB2 Information Integrator Support Web site
Replication and Event Publishing Guide and Reference	SC18-7568	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
SQL Replication Guide and Reference	SC27-1121	DB2 Information Integrator Support Web site
Tuning for Replication and Event Publishing Performance	N/A	DB2 Information Integrator Support Web site
Tuning for SQL Replication Performance	N/A	DB2 Information Integrator Support Web site

Table 13. DB2 Information Integrator documentation about event publishing and replication function on Linux, UNIX, and Windows (continued)

Name	Form number	Location
Release Notes for IBM DB2 Information Integrator Standard Edition, Advanced Edition, and Replication for z/OS	N/A	• In the DB2 Information Center, **Product Overviews** > **Information Integration** > **DB2 Information Integrator overview** > **Problems, workarounds, and documentation updates** • DB2 Information Integrator Installation launchpad • DB2 Information Integrator Support Web site • The *DB2 Information Integrator* product CD

Documentation about federated function on z/OS

Table 14. DB2 Information Integrator documentation about federated function on z/OS

Name	Form number	Location
Client Guide for Classic Federation and Event Publisher for z/OS	SC18-9160	DB2 Information Integrator Support Web site
Data Mapper Guide for Classic Federation and Event Publisher for z/OS	SC18-9163	DB2 Information Integrator Support Web site
Getting Started with Classic Federation for z/OS	GC18-9155	DB2 Information Integrator Support Web site
Installation Guide for Classic Federation and Event Publisher for z/OS	GC18-9301	DB2 Information Integrator Support Web site
Reference for Classic Federation and Event Publisher for z/OS	SC18-9156	DB2 Information Integrator Support Web site
System Messages for Classic Federation and Event Publisher for z/OS	SC18-9162	DB2 Information Integrator Support Web site
Transaction Services Guide for Classic Federation for z/OS	SC18-9161	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Classic Federation for z/OS	N/A	DB2 Information Integrator Support Web site

Documentation about federated function on Linux, UNIX, and Windows

Table 15. DB2 Information Integrator documentation about federated function on Linux, UNIX, and Windows

Name	Form number	Location
Application Developer's Guide	SC18-7359	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site

Table 15. DB2 Information Integrator documentation about federated function on Linux, UNIX, and Windows (continued)

Name	Form number	Location
C++ API Reference for Developing Wrappers	SC18-9172	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Data Source Configuration Guide	N/A	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Federated Systems Guide	SC18-7364	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Guide to Configuring the Content Connector for VeniceBridge	N/A	DB2 Information Integrator Support Web site
Installation Guide for Linux, UNIX, and Windows	GC18-7036	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Java API Reference for Developing Wrappers	SC18-9173	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Migration Guide	SC18-7360	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Wrapper Developer's Guide	SC18-9174	• *DB2 PDF Documentation* CD • DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Standard Edition, Advanced Edition, and Replication for z/OS	N/A	• In the DB2 Information Center, **Product Overviews** > **Information Integration** > **DB2 Information Integrator overview** > **Problems, workarounds, and documentation updates** • DB2 Information Integrator Installation launchpad • DB2 Information Integrator Support Web site • The *DB2 Information Integrator* product CD

Documentation about enterprise search function on Linux, UNIX, and Windows

Table 16. DB2 Information Integrator documentation about enterprise search function on Linux, UNIX, and Windows

Name	Form number	Location
Administering Enterprise Search	SC18-9283	DB2 Information Integrator Support Web site
Installation Guide for Enterprise Search	GC18-9282	DB2 Information Integrator Support Web site
Programming Guide and API Reference for Enterprise Search	SC18-9284	DB2 Information Integrator Support Web site
Release Notes for Enterprise Search	N/A	DB2 Information Integrator Support Web site

Release notes and installation requirements

Release notes provide information that is specific to the release and fix pack level for your product and include the latest corrections to the documentation for each release.

Installation requirements provide information that is specific to the release of your product.

Table 17. DB2 Information Integrator Release Notes and Installation Requirements

Name	File name	Location
Installation Requirements for IBM DB2 Information Integrator Event Publishing Edition, Replication Edition, Standard Edition, Advanced Edition, Advanced Edition Unlimited, Developer Edition, and Replication for z/OS	Prereqs	• The *DB2 Information Integrator* product CD • DB2 Information Integrator Installation Launchpad
Release Notes for IBM DB2 Information Integrator Standard Edition, Advanced Edition, and Replication for z/OS	ReleaseNotes	• In the DB2 Information Center, **Product Overviews** > **Information Integration** > **DB2 Information Integrator overview** > **Problems, workarounds, and documentation updates** • DB2 Information Integrator Installation launchpad • DB2 Information Integrator Support Web site • The *DB2 Information Integrator* product CD
Release Notes for IBM DB2 Information Integrator Event Publisher for IMS for z/OS	N/A	DB2 Information Integrator Support Web site

Table 17. DB2 Information Integrator Release Notes and Installation Requirements (continued)

Name	File name	Location
Release Notes for IBM DB2 Information Integrator Event Publisher for VSAM for z/OS	N/A	DB2 Information Integrator Support Web site
Release Notes for IBM DB2 Information Integrator Classic Federation for z/OS	N/A	DB2 Information Integrator Support Web site
Release Notes for Enterprise Search	N/A	DB2 Information Integrator Support Web site

To view the installation requirements and release notes that are on the product CD:

- On Windows operating systems, enter:

  ```
  x:\doc\%L
  ```

  x is the Windows CD drive letter and *%L* is the locale of the documentation that you want to use, for example, en_US.
- On UNIX operating systems, enter:

  ```
  /cdrom/doc/%L/
  ```

  cdrom refers to the UNIX mount point of the CD and *%L* is the locale of the documentation that you want to use, for example, en_US.

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following list specifies the major accessibility features in DB2® Version 8 products:

- All DB2 functionality is available using the keyboard for navigation instead of the mouse. For more information, see “Keyboard input and navigation.”
- You can customize the size and color of the fonts on DB2 user interfaces. For more information, see “Accessible display.”
- DB2 products support accessibility applications that use the Java™ Accessibility API. For more information, see “Compatibility with assistive technologies” on page 214.
- DB2 documentation is provided in an accessible format. For more information, see “Accessible documentation” on page 214.

Keyboard input and navigation

Keyboard input

You can operate the DB2 tools using only the keyboard. You can use keys or key combinations to perform operations that can also be done using a mouse. Standard operating system keystrokes are used for standard operating system operations.

For more information about using keys or key combinations to perform operations, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard navigation

You can navigate the DB2 tools user interface using keys or key combinations.

For more information about using keys or key combinations to navigate the DB2 Tools, see Keyboard shortcuts and accelerators: Common GUI help.

Keyboard focus

In UNIX® operating systems, the area of the active window where your keystrokes will have an effect is highlighted.

Accessible display

The DB2 tools have features that improve accessibility for users with low vision or other visual impairments. These accessibility enhancements include support for customizable font properties.

Font settings

You can select the color, size, and font for the text in menus and dialog windows, using the Tools Settings notebook.

For more information about specifying font settings, see Changing the fonts for menus and text: Common GUI help.

Non-dependence on color

You do not need to distinguish between colors in order to use any of the functions in this product.

Compatibility with assistive technologies

The DB2 tools interfaces support the Java Accessibility API, which enables you to use screen readers and other assistive technologies with DB2 products.

Accessible documentation

Documentation for DB2 is provided in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

Syntax diagrams are provided in dotted decimal format. This format is available only if you are accessing the online documentation using a screen-reader.

Related concepts:

- "Dotted decimal syntax diagrams" in the *Infrastructure Topics (DB2 Common Files)*

Related tasks:

- "Keyboard shortcuts and accelerators: Common GUI help"
- "Changing the fonts for menus and text: Common GUI help"

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 all 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country/region or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country/region 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 that 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 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. 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 IBM's application programming interfaces.

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.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
AIX
AS/400
DataJoiner
DataPropagator
DB2
DB2 Universal Database
DRDA
Informix
iSeries
MVS
OS/390
OS/400
WebSphere
z/OS
z/VM

The following terms are trademarks or registered trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.

Index

Contacting IBM

To contact IBM customer service in the United States or Canada, call 1-800-IBM-SERV (1-800-426-7378).

To learn about available service options, call one of the following numbers:

- In the United States: 1-888-426-4343
- In Canada: 1-800-465-9600

To locate an IBM office in your country or region, see the IBM Directory of Worldwide Contacts on the Web at www.ibm.com/planetwide.

Product information

Information about DB2 Information Integrator is available by telephone or on the Web.

If you live in the United States, you can call one of the following numbers:

- To order products or to obtain general information: 1-800-IBM-CALL (1-800-426-2255)
- To order publications: 1-800-879-2755

On the Web, go to www.ibm.com/software/data/integration/db2ii/support.html. This site contains the latest information about:

- The technical library
- Ordering books
- Client downloads
- Newsgroups
- Fix packs
- News
- Links to Web resources

Comments on the documentation

Your feedback helps IBM to provide quality information. Please send any comments that you have about this book or other DB2 Information Integrator documentation. You can use any of the following methods to provide comments:

- Send your comments using the online readers' comment form at www.ibm.com/software/data/rcf.
- Send your comments by e-mail to comments@us.ibm.com. Include the name of the product, the version number of the product, and the name and part number of the book (if applicable). If you are commenting on specific text, please include the location of the text (for example, a title, a table number, or a page number).

Printed in USA

SC18-9410-00

Spine information:

IBM DB2 Information Integrator

ASNCLP Program Reference for Replication and Event Publishing

Version 8.2