

IBM<sup>®</sup> DB2 Universal Database<sup>™</sup>



# Administering DB2 Replication: Using The asnclp Program

*Version 8*



IBM<sup>®</sup> DB2 Universal Database<sup>™</sup>



# Administering DB2 Replication: Using The asnclp Program

*Version 8*

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 77.

**First Edition (November 2002)**

| **Last update: October, 2003**

This document is protected by copyright law. The information contained in this publication does not include any product warranties, and any statements provided in this manual should not be interpreted as such.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 2002, 2003. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>About this document</b> . . . . .	<b>v</b>
Who should read this document. . . . .	v
Conventions . . . . .	v
How to send your comments. . . . .	v

<b>Chapter 1. Introduction</b> . . . . .	<b>1</b>
Comparing the asncpl program with other replication interfaces provided by DB2 . . . . .	1
Operating-system environments supported . . . . .	2

<b>Chapter 2. Running the asncpl program</b> . . . . .	<b>3</b>
Setting up your Java environment to run the asncpl program. . . . .	3
Running the administrative commands using the command line interface. . . . .	4
Running the administrative commands using an input file . . . . .	4
Accessing the output files . . . . .	4
The commands . . . . .	5
SET commands . . . . .	5
Task commands . . . . .	5

<b>Chapter 3. The asncpl SET commands</b> . . . . .	<b>7</b>
SET SERVER command. . . . .	8
Syntax . . . . .	9
Parameters . . . . .	9
Usage Notes . . . . .	10
SET PROFILE command . . . . .	10
Syntax . . . . .	10
Parameters . . . . .	11
SET DROP command . . . . .	13
Syntax . . . . .	13
Parameters . . . . .	13
Usage Notes . . . . .	13
SET OUTPUT and SET LOG commands. . . . .	14
Syntax . . . . .	14
Parameters . . . . .	14
Usage notes . . . . .	14
SET CAPTURE SCHEMA command . . . . .	14
Syntax . . . . .	15
Parameters . . . . .	15
SET TRACE command. . . . .	15
Syntax . . . . .	15
Parameters . . . . .	15
SET RUN SCRIPT command . . . . .	15
Syntax . . . . .	16
Parameters . . . . .	16
Usage notes . . . . .	16

<b>Chapter 4. The asncpl control table definition commands</b> . . . . .	<b>17</b>
CREATE CONTROL TABLES command. . . . .	17
Syntax . . . . .	17
Parameters . . . . .	18

Usage Notes . . . . .	19
DROP CONTROL TABLES command. . . . .	19
Syntax . . . . .	20
Parameters . . . . .	20
Usage Notes . . . . .	20

<b>Chapter 5. The asncpl registration definition commands</b> . . . . .	<b>21</b>
CREATE REGISTRATION command . . . . .	21
Syntax . . . . .	21
Parameters . . . . .	22
Usage Notes . . . . .	24
ALTER REGISTRATION command . . . . .	24
Syntax . . . . .	25
Parameters . . . . .	26
Usage Notes . . . . .	27
DROP REGISTRATION command. . . . .	27
Syntax . . . . .	27
Parameters . . . . .	27
Usage Notes . . . . .	27
PROMOTE REGISTRATION command . . . . .	27
Syntax . . . . .	27
Parameters . . . . .	28
Usage Notes . . . . .	28

<b>Chapter 6. The asncpl subscription definition commands</b> . . . . .	<b>31</b>
CREATE SUBSCRIPTION SET command . . . . .	31
Syntax . . . . .	31
Parameters . . . . .	31
Usage Notes . . . . .	32
ALTER SUBSCRIPTION SET command . . . . .	33
Syntax . . . . .	33
Parameters . . . . .	33
CREATE MEMBER command . . . . .	34
Syntax . . . . .	34
Parameters . . . . .	36
Usage Notes . . . . .	41
DROP MEMBER command . . . . .	41
Syntax . . . . .	41
Parameters . . . . .	41
Usage Notes . . . . .	41
ALTER MEMBER ADD COLS command . . . . .	42
Syntax . . . . .	42
Parameters . . . . .	42
Usage Notes . . . . .	42
CREATE STMT command . . . . .	42
Syntax . . . . .	43
Parameters . . . . .	43
DROP STMT command . . . . .	44
Syntax . . . . .	44
Parameters . . . . .	44
Usage Notes . . . . .	44
DROP SUBSCRIPTION SET command . . . . .	44
Syntax . . . . .	44

Parameters . . . . .	44
Usage Notes . . . . .	45
PROMOTE SUBSCRIPTION SET command . . . . .	45
Syntax . . . . .	45
Parameters . . . . .	45
Usage Notes . . . . .	46

<b>Chapter 7. The asncpl offline load utility commands . . . . .</b>	<b>47</b>
OFFLINE LOAD command . . . . .	47
Syntax . . . . .	47
Parameters . . . . .	47

<b>Chapter 8. The asncpl Monitor definition commands . . . . .</b>	<b>49</b>
CREATE CONTACT command . . . . .	49
Syntax . . . . .	49
Parameters . . . . .	49
ALTER CONTACT command . . . . .	50
Syntax . . . . .	50
Parameters . . . . .	50
DROP CONTACT command . . . . .	50
Syntax . . . . .	50
Parameters . . . . .	50
Usage Notes . . . . .	51
SUBSTITUTE CONTACT command . . . . .	51
Syntax . . . . .	51
Parameters . . . . .	51
DELEGATE CONTACT command . . . . .	51
Syntax . . . . .	51
Parameters . . . . .	51
CREATE GROUP command . . . . .	52
Syntax . . . . .	52
Parameters . . . . .	52
ALTER GROUP command . . . . .	52
Syntax . . . . .	52
Parameters . . . . .	52
DROP GROUP command . . . . .	53
Syntax . . . . .	53
Parameters . . . . .	53
Usage Notes . . . . .	53
CREATE ALERT CONDITIONS FOR CAPTURE command . . . . .	53

Syntax . . . . .	53
Parameters . . . . .	53
Usage Notes . . . . .	54
ALTER ALERT CONDITIONS FOR CAPTURE command . . . . .	55
Syntax . . . . .	55
Parameters . . . . .	55
Usage Notes . . . . .	56
DROP ALERT CONDITIONS FOR CAPTURE command . . . . .	56
Syntax . . . . .	56
Parameters . . . . .	56
CREATE ALERT CONDITIONS FOR APPLY command . . . . .	57
Syntax . . . . .	57
Parameters . . . . .	57
Usage Notes . . . . .	58
ALTER ALERT CONDITIONS FOR APPLY command . . . . .	58
Syntax . . . . .	58
Parameters . . . . .	59
Usage Notes . . . . .	60
DROP ALERT CONDITIONS FOR APPLY command . . . . .	60
Syntax . . . . .	60
Parameters . . . . .	60

<b>Chapter 9. Example of running the asncpl program . . . . .</b>	<b>61</b>
Creating databases and tables . . . . .	61
Creating input and output files for the asncpl program . . . . .	61
Input files . . . . .	61
Output files . . . . .	61

<b>Chapter 10. Frequently Asked Questions . . . . .</b>	<b>63</b>
<b>Index . . . . .</b>	<b>67</b>
<b>Notices . . . . .</b>	<b>77</b>
Trademarks . . . . .	79

---

## About this document

DB2 replication provides three interfaces for setting up and maintaining Version 8 replication environments:

- The DB2 Replication Center
- OS/400-system commands
- The **asnclp** program

This document describes the **asnclp** program. This program is a command-line interface for most of the V8 DB2 replication administration tasks. Using the **asnclp** program, you can issue V8 replication administration commands interactively. The output of these commands is always an SQL script, which you can run immediately or at a later time. You can also provide multiple commands in an input file to the **asnclp** program.

For information about the DB2 Replication Center or the OS/400-system commands for replication, see *Replication Guide and Reference* (SC27-1121) or the replication information in the DB2 Information Center.

---

## Who should read this document

This document is written primarily for database administrators and system administrators who must administer and maintain a data replication 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 Universal Database *Replication Guide and Reference* (SC27-1121) and the replication information in the DB2 Information Center.

---

## 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 send your comments

Your feedback helps IBM to provide quality information. Please send any comments that you have about this document or other DB2 DataPropagator documentation. You can use any of the following methods to provide comments:

- Send your comments from the Web. Visit the Web site at:  
[www.ibm.com/software/data/dpropr/](http://www.ibm.com/software/data/dpropr/)

The Web site has a feedback page that you can use to enter and send comments.

- Send your comments by e-mail to [comments@vnet.ibm.com](mailto:comments@vnet.ibm.com). Be sure to 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 chapter and section title, a table number, a page number, or a help topic title).



---

## Chapter 1. Introduction

This chapter presents an overview of the **asnclp** program and contains the following sections:

- “Comparing the **asnclp** program with other replication interfaces provided by DB2”
- “Operating-system environments supported” on page 2

---

### Comparing the **asnclp** program with other replication interfaces provided by DB2

You can use the V8 Replication Center to perform similar tasks to those that you can perform using the **asnclp** program. However, if you plan to automate various replication administration tasks, you need to use a command-line interface. Most users will use the Replication Center to set up their replication environments, but advanced users who rely on automated procedures will use the command-line interface for such tasks.

DB2 DataPropagator for iSeries provides OS/400-native replication administration commands for Version 8. 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 provided by both sets of commands is similar, except for some OS/400-specific parameters.

For DB2 Version 7, the DB2 DataJoiner Replication Administration (DJRA) tool provided a set of administration commands to set up DB2 replication. These commands are not supported in DB2 Version 8 and are replaced by the **asnclp** program. The V7 DJRA commands consisted of:

- Create control tables, drop control tables
- Create table registration, create view registration, alter registration, delete registration, promote registration
- Add set, add member, add statement, alter member (add col), delete set, delete member, delete statement
- After/before load processing
- Promote table, promote table space, promote subscription

All of these DJRA commands are supported by the **asnclp** program, except the “promote table” and “promote table space” commands. 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 supported

The **asnclp** administrative commands are native to the Windows and UNIX operating-system environments only. There is no native support for z/OS or OS/400; that is, 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 supported by the DB2 V8 replication products: z/OS, OS/390, OS/400, 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 Version 8 replication architecture. There is some redundancy in the functionality provided by these two sets of commands for subscription tasks because the DB2 DataPropagator for iSeries provides its own set of OS/400-native commands for replication.

---

## Chapter 2. Running the asncpl program

You can run replication commands either directly from the **asncpl** command line or from an input file. In either case, the **asncpl** program generates output files if at least one command is successful. .

This chapter contains the following sections:

- “Setting up your Java environment to run the asncpl program”
- “Running the administrative commands using the command line interface” on page 4
- “Running the administrative commands using an input file” on page 4
- “Accessing the output files” on page 4
- “The commands” on page 5
- “SET commands” on page 5
- “Task commands” on page 5

---

### Setting up your Java environment to run the asncpl program

Because the **asncpl** program runs in a Java Runtime Environment, you must add the following statements to your Java CLASSPATH environment variable:

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

where *INSTDIR* is the directory in which you installed DB2.

#### Examples:

For Windows:

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

For UNIX:

```
export CLASSPATH=$CLASSPATH:/usr/bin/sqllib/java/Common.jar:/usr/bin/  
sqllib/tools/db2replapis.jar:/usr/bin/sqllib/tools/db2cmn.jar:/usr/bin/  
sqllib/tools/jt400.jar:
```

**Notes:** To run the **asncpl** program with remote z/OS servers, you must bind the local DB2 database with the basic DRDA and CLI packages:

```
db2 bind @ddcmvs.lst isolation ur blocking all  
db2 bind @db2cli.lst isolation ur blocking all
```

If you do not perform this bind, the first time you run the **asncpl** program with a remote z/OS server, the **asncpl** program will issue 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 administrative commands using the command line interface

To run the **asnclp** program interactively, use the **asnclp** command 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 replication commands described in “The commands” on page 5.

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

```
Repl > quit
```

From the operating-system command prompt, you can receive 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 administrative commands using an input file

To run the **asnclp** program in a batch mode using an input file, use the **asnclp** command with `-f` and the input-file name as parameters. 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 instead of a 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.

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 either a bad command or the end of the file.

---

## Accessing the output files

The output from the replication commands is always directed to files. These files are specified by the **SET OUTPUT** and **SET LOG** commands. The output for the log file is also directed to the operating system’s standard output (stdout).

The **asnclp** program typically generates the following files:

- Log file

- SQL script file or files, if at least one replication command completed successfully

---

## The commands

There are two general classes of commands for the **asnclp** program: environmental setup (**SET**) commands and **task** commands.

### SET commands

The environmental setup commands provide a common environment for all subsequent replication commands. These commands are also known as the **SET** commands and consist of:

- **SET SERVER**
- **SET PROFILE**
- **SET DROP**
- **SETOUTPUT, SET LOG**
- **SET CAPTURE SCHEMA**
- **SET TRACE**
- **SET RUN TRACE**

The scope of these commands is limited to a single **asnclp** command-line session or to a single input file.

A **SET** command affects *all* of the subsequent **task** commands.

You can specify multiple **SET** commands for an **asnclp** command-line session. You can also specify the same **SET** command several times in a command line session; in this case, the most recent **SET** command overrides any previous **SET** commands.

### Task commands

The **task** commands provide the functionality for setting up your replication environment and consist of:

- **CREATE CONTROL TABLES**
- **DROP CONTROL TABLES**
- **CREATE REGISTRATION**
- **ALTER REGISTRATION**
- **DROP REGISTRATION**
- **PROMOTE REGISTRATION**
- **CREATE SUBSCRIPTION SET**
- **ALTER SUBSCRIPTION SET**
- **CREATE MEMBER**
- **DROP MEMBER**
- **ALTER MEMBER ADD COLS**
- **CREATE STMT**
- **DROP STMT**
- **DROP SUBSCRIPTION SET**
- **PROMOTE SUBSCRIPTION SET**
- **OFFLINE LOAD**

- CREATE CONTACT
- ALTER CONTACT
- DROP CONTACT
- SUBSTITUTE CONTACT
- DELEGATE CONTACT
- CREATE GROUP
- ALTER GROUP
- DROP GROUP
- CREATE ALERT CONDITIONS FOR CAPTURE
- ALTER ALERT CONDITIONS FOR CAPTURE
- DROP ALERT CONDITIONS FOR CAPTURE
- CREATE ALERT CONDITIONS FOR APPLY
- ALTER ALERT CONDITIONS FOR APPLY
- DROP ALERT CONDITIONS FOR APPLY

---

## Chapter 3. The `asncpl` SET commands

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 `asncpl` SET commands include:

- "SET SERVER command" on page 8
- "SET PROFILE command" on page 10
- "SET DROP command" on page 13
- "SET OUTPUT and SET LOG commands" on page 14
- "SET CAPTURE SCHEMA command" on page 14
- "SET TRACE command" on page 15
- "SET RUN SCRIPT command" on page 15

These SET commands define a the following environmental 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, when you set a Capture control server to a particular database alias, all subsequent task commands use that Capture server alias so that all CREATE REGISTRATION commands run against that database.

Running the SET SERVER command is mandatory before running any task command.

**Note:** You cannot simply use the DB2 environment variable DB2DBDFT because the `asncpl` program often needs to connect to more than one server, such as in the case of creating subscription-set members.

### Optional customization for table spaces and indexes

The "Edit Logic" exit routines of DJRA allowed you to code your own customization logic into the DJRA program. These exit routines were mainly used for:

1. Customization of CREATE TABLESPACE and CREATE INDEX statements.
2. Customization of names for new replication objects: CD and CCD tables, target tables, and so on, including platform differences for z/OS or Windows, for example.

To address the need to be able to customize 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.

To address the need to be able to customize object names, the underlying replication API uses a good default naming convention so that no **SET** command is necessary.

**Differences with DJRA Edit Logic:** You must issue the **asnclp SET** 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 **asnclp SET** 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.

#### **Drop information**

The **SET DROP** command allows you to set the **DROP** environment variables for tables and table spaces. These changes will 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.

**Note:** Currently, there is no support for a **SET INPUT** type command that would allow you to import a file that contains many **SET** commands that could be reused across sessions.

#### **Capture schema**

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

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

#### **Run option**

The **SET RUN SCRIPT** command allows you to run replication SQL scripts automatically after they are generated.

---

## **SET SERVER command**

The **SET SERVER** command assigns a database alias for a logical replication server (Remote Source server, Capture control server, Apply control server, target server, or Monitor control 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**

You must always set the Capture control server, Apply control server, or Monitor Control server before creating or dropping replication control tables.

**Note:** If a target server is also a Capture control server, you must assign the Capture server variable to the target server alias name.

#### **All registration commands (including promote)**

You must always set the Capture control server before running the registration commands. For iSeries, the Remote Source server must also be set.



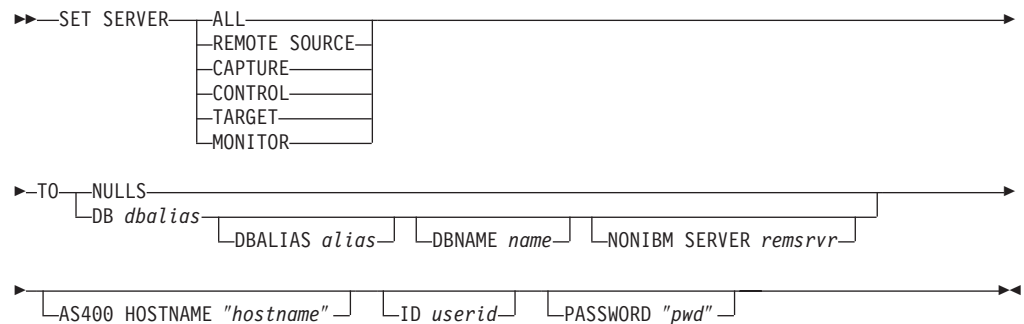
### All subscription commands (including promote)

You should 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 or Monitor control servers for these commands. For iSeries, you must set the Remote Source server.

### All Monitor administration commands

You should always set the Monitor control server before running the monitor administration commands.

## Syntax



## Parameters

### ALL, REMOTE SOURCE, CAPTURE, CONTROL, TARGET, MONITOR

Specifies which replication logical server to associate with the database alias provided:

- All servers (Remote source, Capture control, Apply control, target, and Monitor control)
- Remote Source server
- Capture control server
- Apply control server
- Target server
- Monitor control server

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

### NULLS

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

### DB, DBALIAS

Specifies the DB2 alias name.

### DBNAME

Specifies the DB2 database 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 or Monitor 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 REMOTE SERVER clause, the **asnclp** program calls the heterogeneous replication API; otherwise, the **ansclp** program calls DB2 replication APIs. "Heterogeneous" implies non-IBM data sources such as Oracle and Sybase. The heterogeneous APIs support replication to and from non-IBM data sources. The **SET** command saves the database server information, but does not perform the actual db2 connect statement. The **SET** 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 **SET** commands, the most recent command overrides the current settings for a given Remote Source, Capture control, Apply control, Monitor control, or target server. In other words, 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 DB2 tablespace object. After you issue a **SET PROFILE** command, all subsequent **task** commands inherit the tablespace 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 the remote server.

For OS/400 systems, there are no 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).

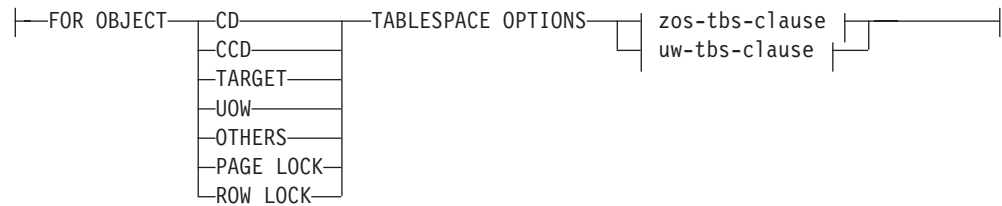
## Syntax

```

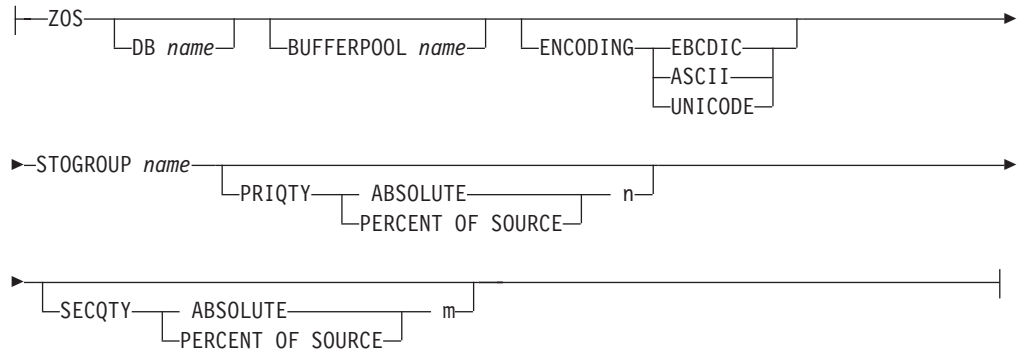
▶▶ SET PROFILE name [ prof-clause ]
                    [ UNDO ]
▶▶────────────────────────────────────────────────────────────────────────────────▶▶

```

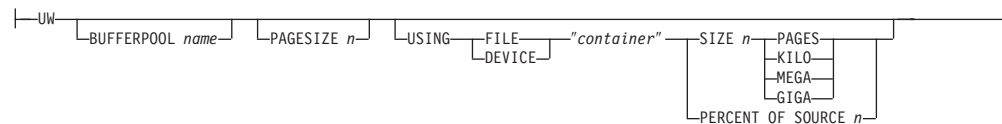
### prof-clause:



### zos-tbs-clause:



### uw-tbs-clause:



## 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 tables, except 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 it might be needed is for the target table, but the **SET** 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 z/OS database name to connect to. 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 bufferpool 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 a percentage value.

## 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

Specifies that the profile name and all set values to should be NULL.

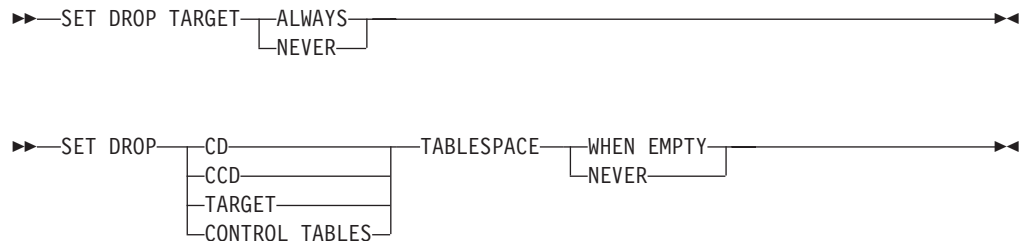
---

## 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 **SET**-command level), whereas the create options are at an object level (that is, they are at the **task**-command level).

### Syntax



### Parameters

#### TARGET

Specifies that the target table be dropped if a subscription-set member is dropped or if an entire subscription set is dropped.

#### CD, CCD, TARGET, CONTROL TABLES, TABLESPACE

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.

### Usage Notes

- WHEN EMPTY relates to anything in the table space.
- 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 TABLE** command allows it.

---

## SET OUTPUT and SET LOG commands

The **SET OUTPUT** and **SET LOG** commands allow you to define output files for the replication command line interface. The output files contain the administration SQL statements needed to set up replication. The log file contains informational messages, warnings, and errors.

### Syntax

```
▶▶ SET OUTPUT [CAPTURE SCRIPT "capfname" ] [CONTROL SCRIPT "cntlfname" ]
▶▶ [TARGET SCRIPT "trgfname" ] [MONITOR SCRIPT "monfname" ]
▶▶ SET LOG "logfname"
```

### Parameters

#### CAPTURE SCRIPT

Specifies the output file name for scripts to be executed at the Capture server. The default file name is replcap.sql.

#### CONTROL SCRIPT

Specifies the output file name for scripts to be executed at the Apply control server. The default file name is replctl.sql.

#### TARGET SCRIPT

Specifies the output file name for scripts to be executed at the target server. The default file name is repltrg.sql.

#### MONITOR SCRIPT

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

#### LOG NAME

Specifies the log file name. The default file name is replmsg.log.

### 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.

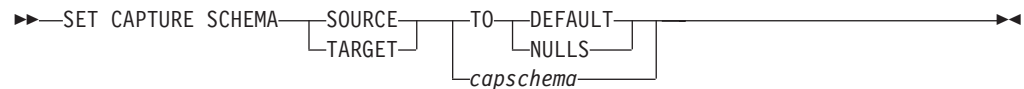
---

## SET 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 users of the **task** commands to ignore the Capture schema settings.

## Syntax



## 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` commands will be reset.

### 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 (normal trace) and stderr (stacktraces).

## Syntax



## Parameters

**ON** Turns on the trace.

**OFF** Turns off the trace.

---

## 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 `asnclp` commands process the next `task` command.

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 based on the minimum

synchpoint. This trigger is dropped and re-created 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, the prune control trigger in the database does not have the CCD information for the previous registration, and the trigger will be out of synch 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



## 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

**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 across servers.

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

## Usage notes

- This command supports heterogeneous replication scripts.
- If the original **task** command (for example, the **CREATE REGISTRATION** command) fails due to errors, no SQL script is generated, so no SQL statements are executed.



---

## Chapter 4. The asncip control table definition commands

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 19

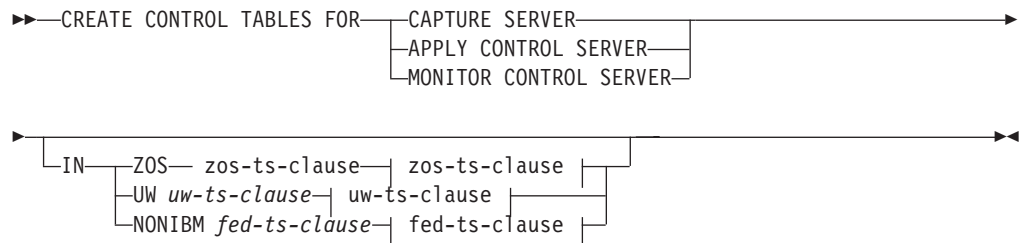
---

### CREATE CONTROL TABLES command

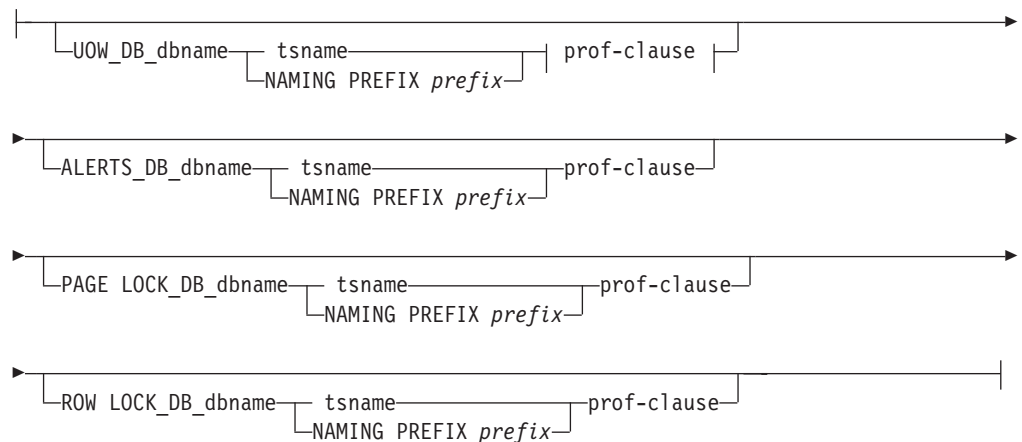
The **CREATE CONTROL TABLES** command creates a new set of Capture, Apply, or Monitor control tables.

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

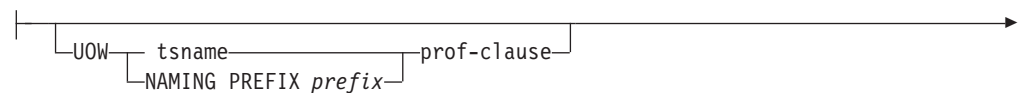
#### Syntax

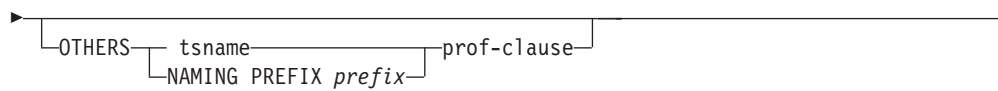


#### zos-ts-clause:

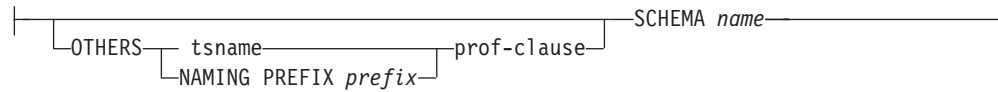


#### uw-ts-clause:





**fed-ts-clause:**



**prof-clause:**



## Parameters

**CAPTURE SERVER, APPLY CONTROL SERVER, MONITOR CONTROL SERVER**

Specifies the logical server to create replication control tables for.

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

- The **CREATE USING PROFILE** clause must be specified before before you can use the **REUSE** clause.
- If the **CREATE USING PROFILE** clause 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** clause 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** clause 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** 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 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 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 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.
- The double quotation marks are mandatory

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

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

**ALERTS**

Specifies the table space for the monitor alerts table. This keyword is valid only for Monitor control servers.

**PAGE LOCK**

Specifies the table space for those replication control tables that require page-level locking.

**ROW LOCK**

Specifies the table space for those replication control tables that require row-level locking.

**DB** Specifies the database name.

**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.

## Usage Notes

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

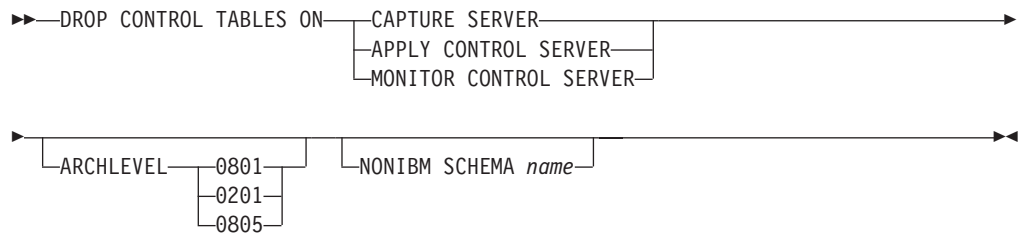
---

## DROP CONTROL TABLES command

The **DROP CONTROL TABLES** command drops a set of Capture, Apply, or Monitor control tables. In addition to dropping Version 8 replication control tables, you can use this command to drop Version 7 (or earlier) replication control tables. Being able to drop pre-V8 control tables can be useful for those environments that included empty replication control tables because the GUI created all control tables in each server, regardless of how you used that server.

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

## Syntax



## 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. The default is 0801, which specifies the Version 8 architecture level. 0201 specifies the architecture level for Version 5, Version 6, or Version 7. For the Monitor control tables, the architecture level is always 0801.

### **NONIBM SCHEMA**

Specifies the remote schema name to use for heterogeneous replication.

## Usage Notes

- The **SET DROP** command affects this command.
- **Recommendation:** Migrate the pre-V8 replication control tables instead of dropping them if the tables contain any data.

---

## Chapter 5. The asncip registration definition commands

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

The **registration definition** commands include:

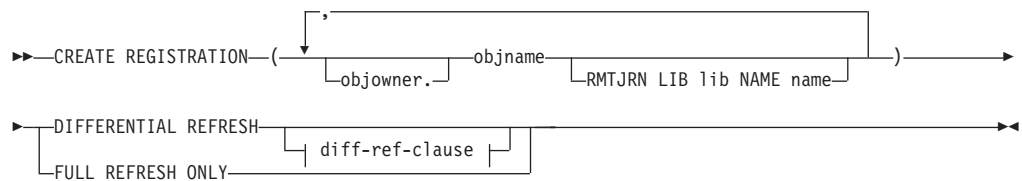
- “CREATE REGISTRATION command”
- “ALTER REGISTRATION command” on page 24
- “DROP REGISTRATION command” on page 27
- “PROMOTE REGISTRATION command” on page 27

---

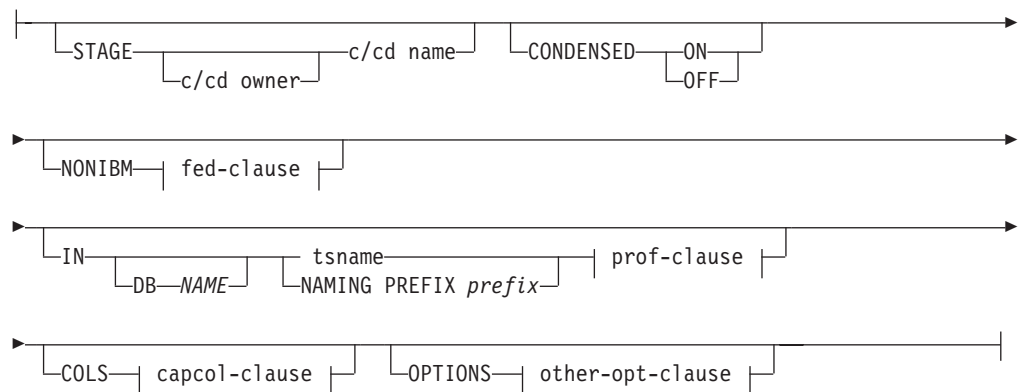
### CREATE REGISTRATION command

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.

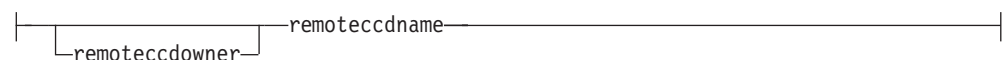
#### Syntax



#### diff-ref-clause:



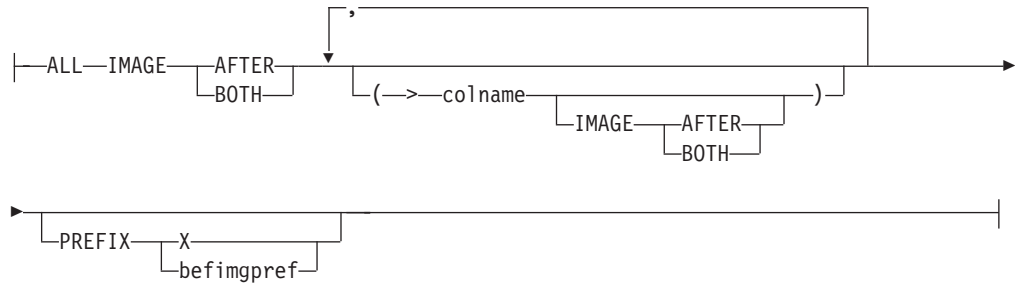
#### fed-clause:



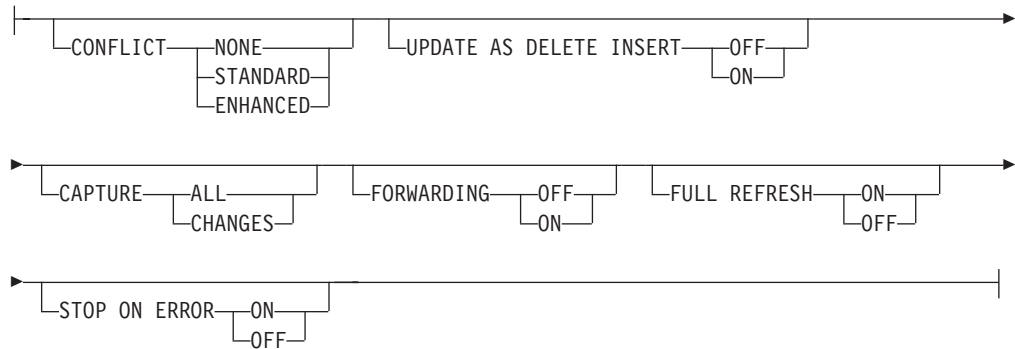
### prof-clause:



### capcol-clause:



### other-opt-clause:



## Parameters

### objowner, objname

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

### RMTJRN

LIB (the AS400 library name); NAME (the AS400 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.

This parameter is ignored for a CD table because a CD table is always noncondensed.

**FULL REFRESH ONLY**

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

**NONIBM**

Specifies the non-IBM options.

**remote ccd owner, remote ccd name**

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.

**COL** 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**

Specifies the before-image prefix. The default is X.

**Note:** The before-image prefix can be the null string. In this case, the replication API will insert the prefix as a NULL value in the IBMSNAP\_REGISTER table. If the prefix is null, the registered source cannot allow any before-image columns, and you cannot alter the registration to change it.

#### OPTIONS

Specifies other registration options.

#### CONFLICT

Specifies the conflict-detection level. The default is NONE.

#### 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

**ON** Specifies that changes from this source are forwarded.

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

#### 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 once:

- 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.

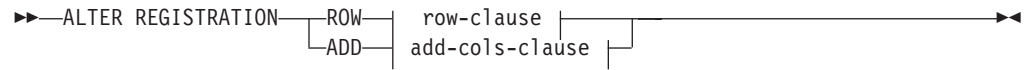
---

## ALTER REGISTRATION command

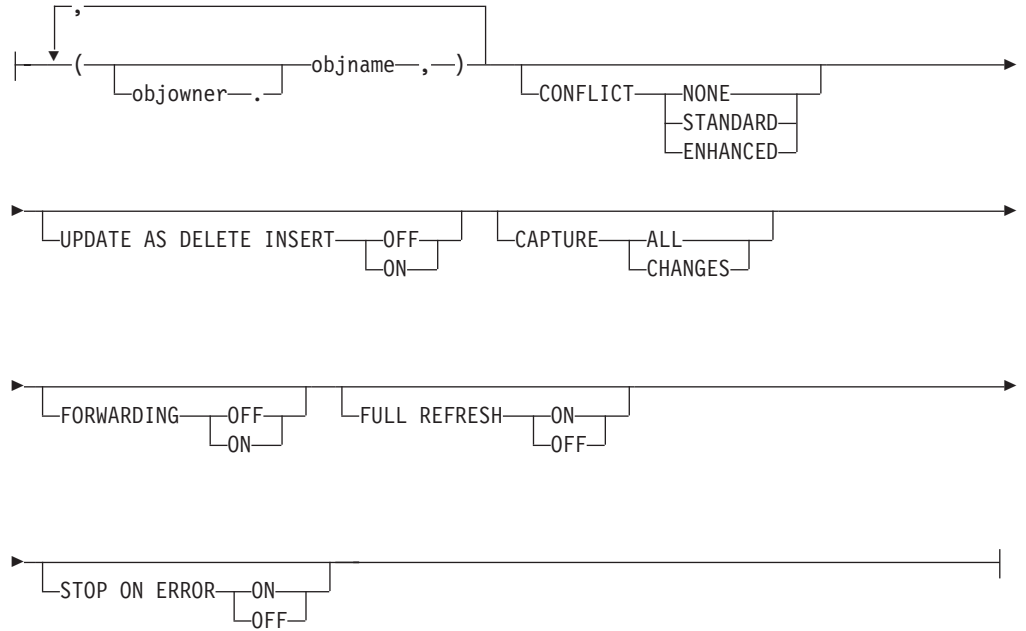
The ALTER REGISTRATION command alters a registration row in the IBMSNAP\_REGISTER table and allows you to add new columns to a registered source.



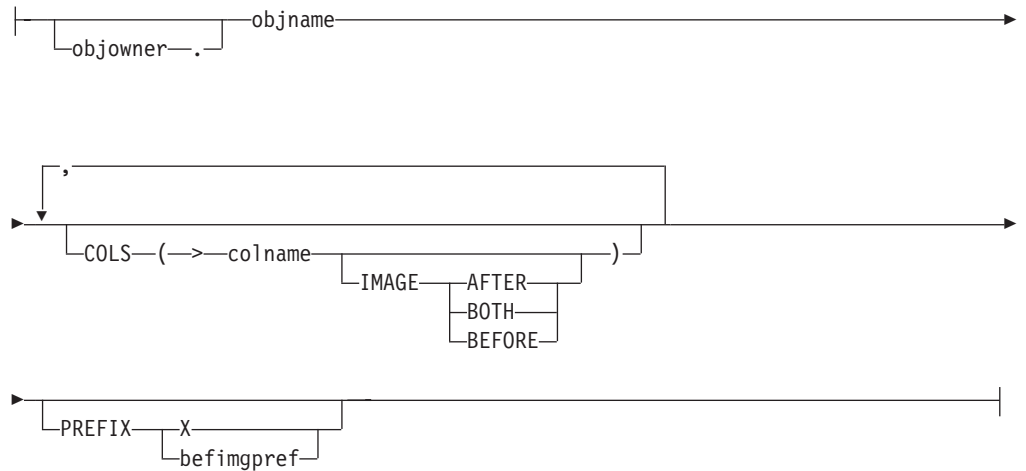
# Syntax



## row-clause:



## add-cols-clause:



## Parameters

- ROW** Specifies the row identified by object owner and object name. This row is 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 object owner and object name. 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**  
Specifies the before-image prefix.
- Note:** The before-image prefix can be the null string. In this case, the replication API will insert the prefix as a NULL value in the IBMSNAP\_REGISTER table. If the prefix is null, the registered source cannot allow any before-image columns, and you cannot alter the registration to change it.

## 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:

- **ALTER ADD REGISTRATION COL** to add column to CD table
- **ALTER ADD SUBSCRIPTION MEMBER COL** 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 ( [objowner.] objname [, -] )
```

### 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 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 ( [objowner.] objname [, -] ) [USING new-clause]
```

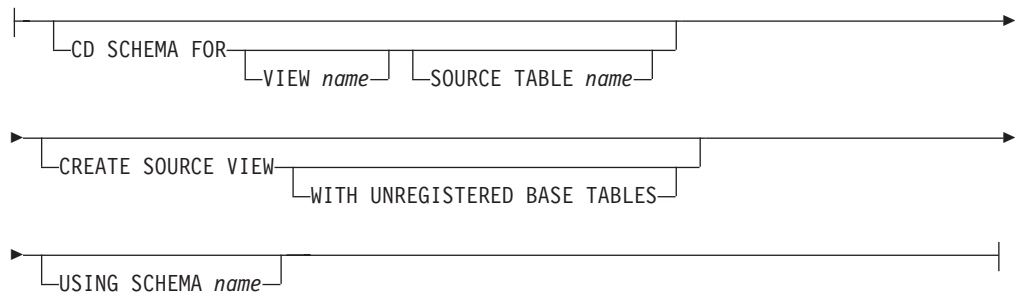
**new-clause:**

```
[SOURCE DB alias] [CAPTURE SCHEMA name] [TABLE | VIEW] [tbl-clause | view-clause]
```

**tbl-clause:**

```
[CD SCHEMA name] [CREATE SOURCE WITH SCHEMA name]
```

### view-clause:



## 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**

Specifies the new Capture schema for the promoted object.

### **CD SCHEMA**

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.

**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.

### **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 have already been 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.

- 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 you should not change the CD schema during when promoting registrations.



---

## Chapter 6. The asncip subscription definition commands

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

The **subscription definition** commands include:

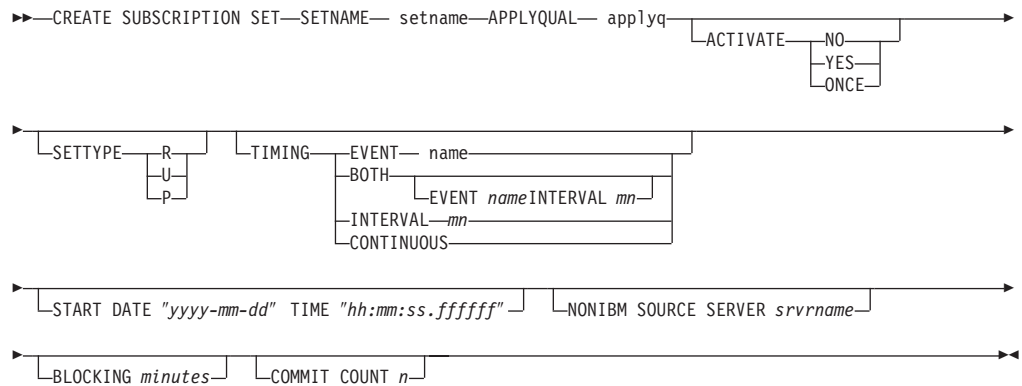
- “CREATE SUBSCRIPTION SET command”
- “ALTER SUBSCRIPTION SET command” on page 33
- “CREATE MEMBER command” on page 34
- “DROP MEMBER command” on page 41
- “ALTER MEMBER ADD COLS command” on page 42
- “CREATE STMT command” on page 42
- “DROP STMT command” on page 44
- “DROP SUBSCRIPTION SET command” on page 44
- “PROMOTE SUBSCRIPTION SET command” on page 45

---

### CREATE SUBSCRIPTION SET command

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

#### Syntax



#### 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 should be deactivated. This is the default.

**YES** Specifies that the subscription set should be activated.

**ONCE** Specifies that the subscription set should 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.

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

##### **INTERVAL**

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

##### **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**

Specifies the number of transactions that the **Apply** program should process before issuing an 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 has processed the entire set.

## **Usage Notes**

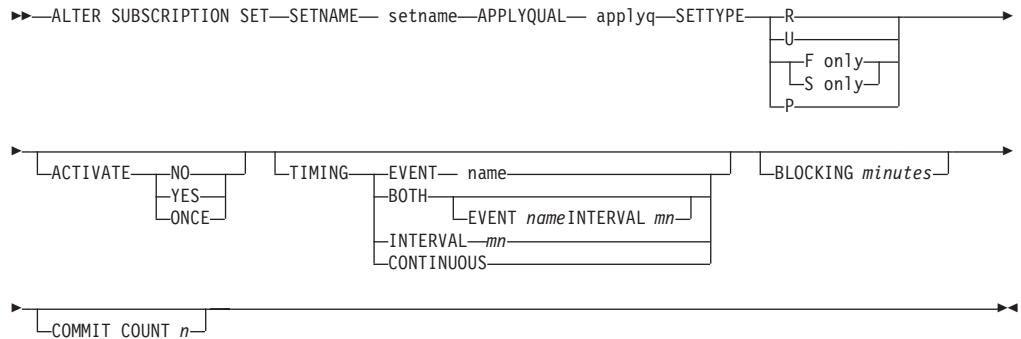
- This command can create only empty subscription sets, whereas the Replication Center allows you to create empty subscription sets or add members to the set while creating it.
- A Capture schema is required, even though the set is empty.
- Because the set is empty, the default for activating the set is **NO**.
- To add a 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



### 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 should be deactivated.

**YES** Specifies that the subscription set should be activated.

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

#### SETTYPE

Specifies the subscription-set type.

**R** Specifies a read-only set.

**U** Specifies an update-anywhere set.

**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.

#### 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.

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

### INTERVAL

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

### 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. This keyword controls the MAX\_SYNC\_MINUTES column of the IBMSNAP\_SUB\_SET table.

### COMMIT COUNT

Specifies the number of transactions that the **Apply** program should process before issuing an SQL COMMIT statement for the subscription set.

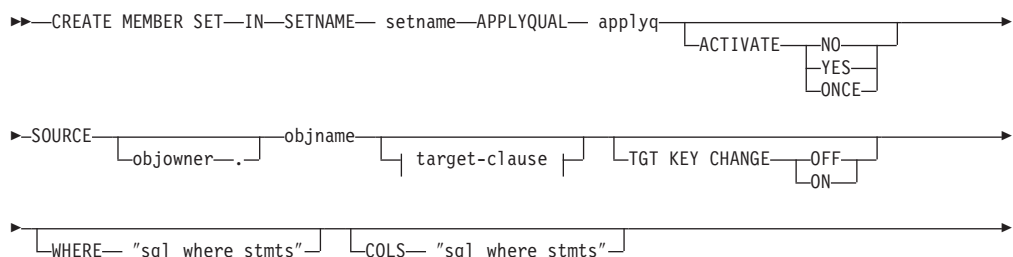
---

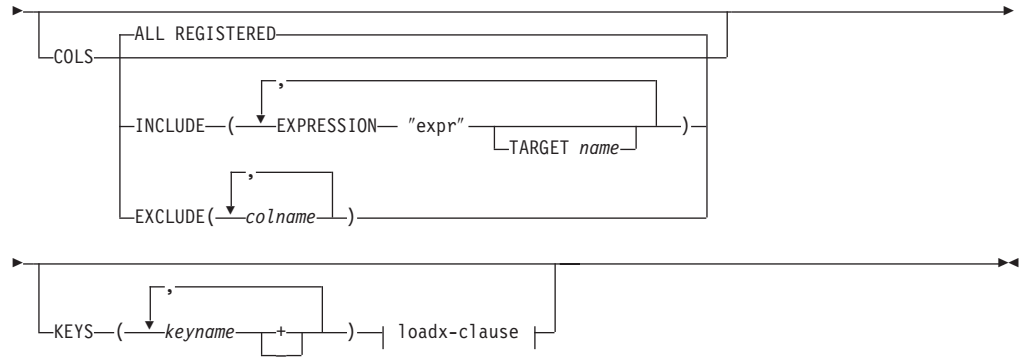
## CREATE MEMBER command

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

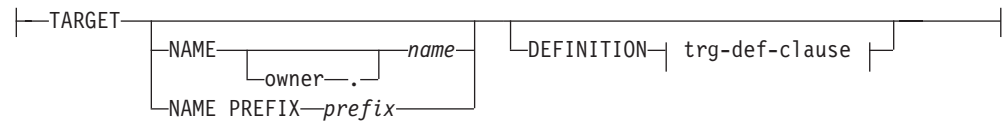
- Creating the mapping between the source and target tables (database objects).
- Creating the mapping between the source and target columns.
- Creating the target table (database object), if it doesn't already exist.
- Creating the target index, if necessary.
- Setting the IS\_KEY value for the index.

### Syntax

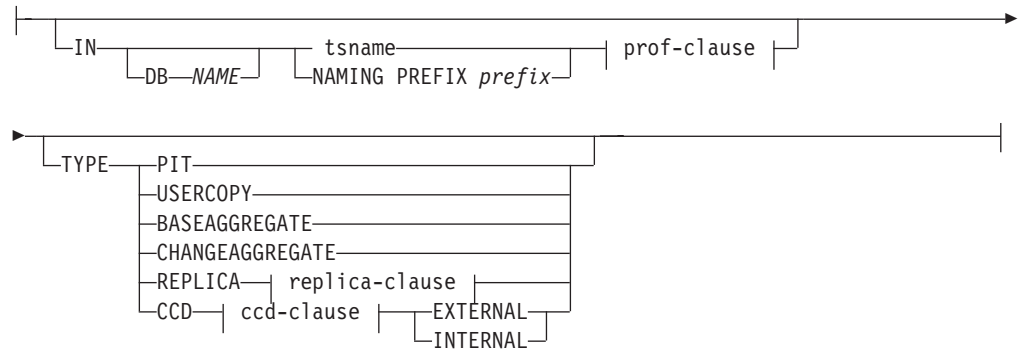




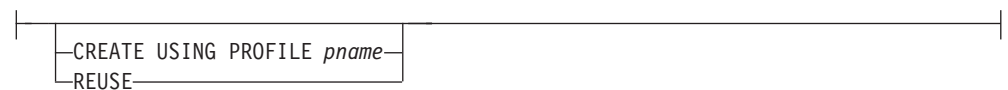
**target-clause:**



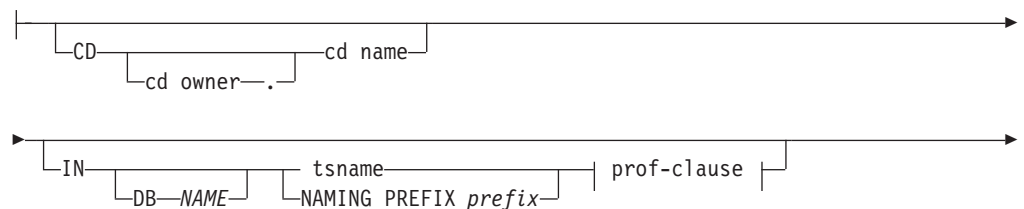
**trg-def-clause:**

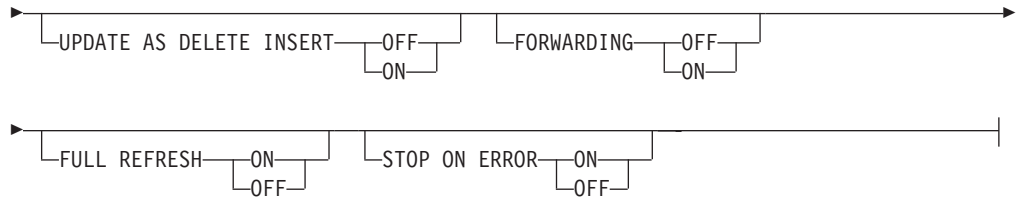


**prof-clause:**

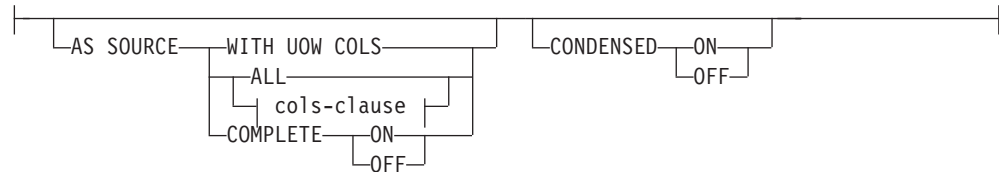


**replica-clause:**





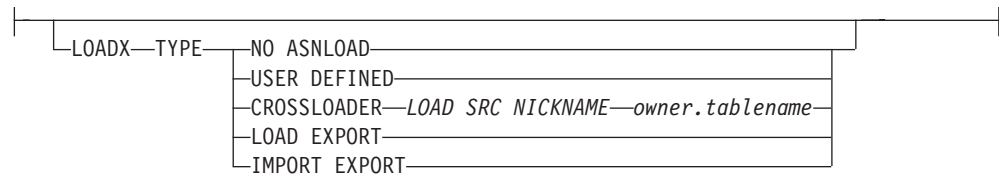
**ccd-clause:**



**cols-clause:**



**loadx-clause:**



**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 should be deactivated. This is the default.

**YES** Specifies that the subscription set should be activated.

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

**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 Table Name are required.

**LOAD EXPORT**

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

**IMPORT EXPORT**

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

**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*:
  - 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 tablespace 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 *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*:
  - 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 table).

#### **REPLICA**

Specifies a replica table for update-anywhere replication.

**CD** Specifies cowner 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*:
  - 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 tablespace 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 *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*:
  - 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 full refresh for the replica table.

- ON** Specifies that full refresh should be performed. This is the default.
- OFF** Specifies that full refresh should be performed.

#### STOP ON ERROR

Specifies whether the Capture program should stop when encountering an error.

- ON** Specifies that the Capture program should stop if a Capture error occurs. This is the default.
  - OFF** Specifies that the Capture program should 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**  
Specifies that the CCD table should include columns from the IBMSNAP\_UOW table.
    - ALL** Specifies that the CCD table should include all UOW columns.
    - colname**  
Specifies the specific columns from the UOW table that the CCD table should include. 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 should include all data. This is the default.
    - OFF** Specifies that the CCD table should include only changes.
  - CONDENSED**  
Specifies whether the CCD table is condensed.
    - ON** Specifies that the CCD table should include only the most recent change for each row. This is the default.
    - OFF** Specifies that the CCD table should include a change history for each row.
  - EXTERNAL**  
Specifies that the CCD table is external.
  - INTERNAL**  
Specifies that the CCD table is internal.
- TGT KEY CHANGE**  
Specifies whether the target key can change.
  - ON** The key can change.
  - OFF** The key 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**

Include the specified columns

**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, it 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—FROM—SETNAME— setname—APPLYQUAL— applyq—SOURCE—▶
▶—objowner—..]—objname—TARGET—objname—▶
  [objowner—..]
  [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** command.

- Whether the target table is also dropped depends on the **SET** command. However, 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—IN—SETNAME— setname—APPLYQUAL— applyq—SOURCE—▶▶
▶ [objowner—..]—objname—TARGET— [objowner—..]—objname—▶▶
▶COLS—(—|—EXPRESSION "source col/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

Specifies the column name for the target.

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

### Usage Notes

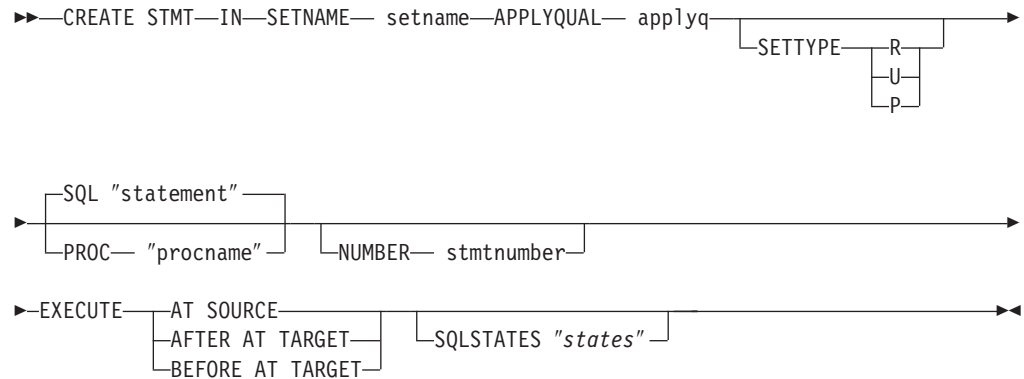
- For update-anywhere subscription sets, the columns are added to the members for both replication directions (master-to-replica and replica-to-master).
- The Capture schema for the target table is inherited from the subscription set.

---

## CREATE STMT command

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

## Syntax



## 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

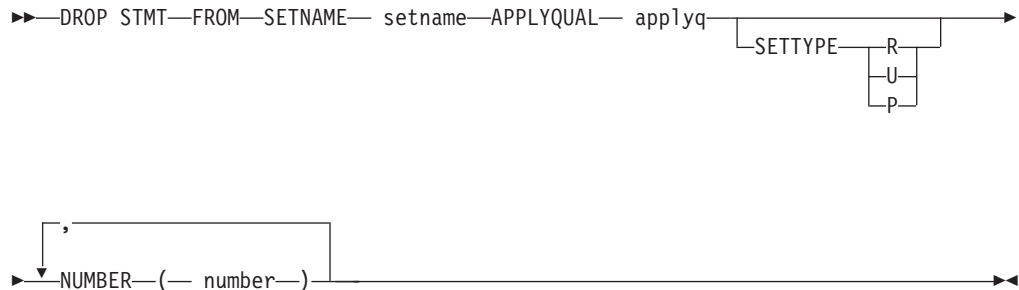
Specifies the SQL states that are accepted as normal during execution of the statement or procedure. The double quotation marks are required.

---

## DROP STMT command

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

### Syntax



### 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



### 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 41 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 SETNAME setname APPLYQUAL applyq [USING new clause]
```

### new-clause:

```
►► [CAPTURE SCHEMA FOR [SOURCE name] [REPLICA name]]
```

```
►► [DB FOR [SOURCE alias] [TARGET alias] [CONTROL alias]]
```

```
►► [APPLYQUAL name] [SETNAME name]
```

```
►► [SOURCE SCHEMA name TARGET [SCHEMA name CD 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 during when promoting registrations.

---

## Chapter 7. The asncip offline load utility commands

The **offline load utility** 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.

The **offline load utility** command is:

- “OFFLINE LOAD command”

---

### OFFLINE LOAD command

The **OFFLINE LOAD** command allows you to control a manual full refresh for offline load procedures.

#### Syntax

```
▶▶—OFFLINE LOAD—

|        |
|--------|
| BEFORE |
| AFTER  |

—SETNAME setname—APPLYQUAL— applyq—▶▶
```

#### 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

Specifies the subscription-set name.

##### APPLYQUAL

Specifies the Apply qualifier for the subscription set.





---

## Chapter 8. The asncip Monitor definition commands

The **Monitor definition** commands are **task** commands that executes within the context of the replication command-line interface. They inherit the context defined by **SET SERVER** command. Use the **control table definition** commands to create or drop the Monitor control tables.

The **Monitor definition** commands include:

- “CREATE CONTACT command”
- “ALTER CONTACT command” on page 50
- “DROP CONTACT command” on page 50
- “SUBSTITUTE CONTACT command” on page 51
- “DELEGATE CONTACT command” on page 51
- “CREATE GROUP command” on page 52
- “ALTER GROUP command” on page 52
- “DROP GROUP command” on page 53
- “CREATE ALERT CONDITIONS FOR CAPTURE command” on page 53
- “ALTER ALERT CONDITIONS FOR CAPTURE command” on page 55
- “DROP ALERT CONDITIONS FOR CAPTURE command” on page 56
- “CREATE ALERT CONDITIONS FOR APPLY command” on page 57
- “ALTER ALERT CONDITIONS FOR APPLY command” on page 58
- “DROP ALERT CONDITIONS FOR APPLY command” on page 60

---

### CREATE CONTACT command

The **CREATE CONTACT** command creates contact information, such as the contact name and email address, that the Replication Alert Monitor program uses for notifications when a replication alert condition is detected. You can optionally associate a contact to a pre-existing group.

#### Syntax

```
►► CREATE CONTACT contact-name [GROUP group-name] [EMAIL PAGE]
► "email-address" [DESCRIPTION "desc"]
```

#### Parameters

##### CONTACT

Specifies the name of the contact. This name cannot match another contact already defined. This parameter is required.

##### GROUP

Specifies the name of the group. The group must be already defined.

**EMAIL**

Specifies the primary email address for the contact. The double quotation marks are required.

**PAGE**

Specifies the pager address for the contact. The double quotation marks are required.

**SUBSCRIPTION**

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 email address, that the Replication Alert Monitor program uses for notifications when a replication alert condition is detected.

**Syntax**

```

▶▶ALTER CONTACT—contact-name—┌EMAIL┐—"email-address"—┌DESCRIPTION—"desc"—┐
└PAGE└

```

**Parameters****CONTACT**

Specifies the name of the contact. The contact must exist. This parameter is required.

**EMAIL**

Specifies the primary email address for the contact. The double quotation marks are required.

**PAGE**

Specifies the pager address for the contact. The double quotation marks are required.

**SUBSCRIPTION**

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—contact-name—┌SUBSTITUTE WITH— contact-name┐
└└

```

**Parameters****CONTACT**

Specifies the name of the contact. The contact must exist. This parameter is required.

### SUBSTITUTE WITH

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 to 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—contact-name1— WITH— contact-name2—►►
```

### Parameters

#### CONTACT

Specifies the name of the contact to be substituted. The contact must exist. This parameter is required.

**WITH** Specifies the new contact for all alert conditions (if any) that refer to the contact being substituted. The 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—contact-name1— TO— contact-name2—FROM—"start-date" —►►  
►—TO—"end-date" —►►
```

### Parameters

#### CONTACT

Specifies the name of the contact to be delegated. The contact must exist. This parameter is required.

**TO** 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

Specifies the date when the delegation starts. The date is sensitive to the DB2 locale. The double quotation marks are required.

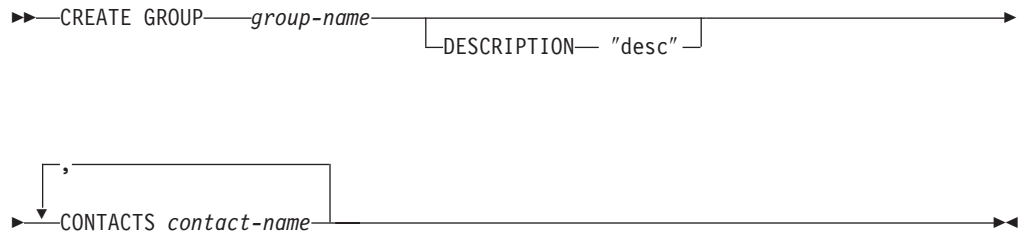
**TO** Specifies the date when the delegation ends. The date is sensitive to the DB2 locale. The double quotation marks are required.

---

## CREATE GROUP command

The **CREATE GROUP** command creates a group of replication monitor contacts.

### Syntax



### Parameters

#### GROUP

Specifies the name of the group. This name cannot match another group already defined. This parameter is required.

#### DESCRIPTION

Specifies a brief description for the group. The double quotation marks are required.

#### CONTACTS

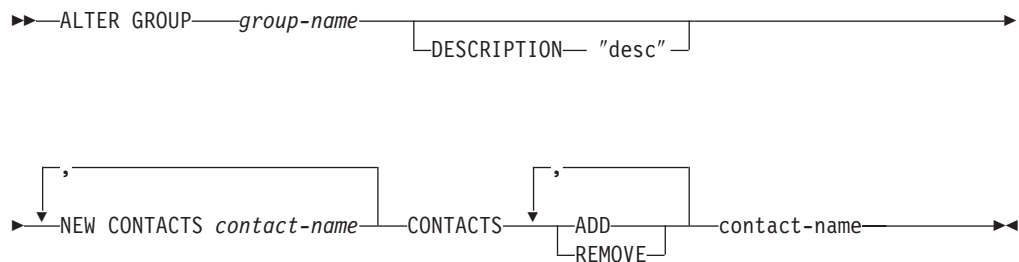
Specifies a comma-separated list of contacts that belong to this group.

---

## ALTER GROUP command

The **ALTER GROUP** command alters a group of replication monitor contacts.

### Syntax



### Parameters

#### GROUP

Specifies the name of the group. The group must exist. This parameter is required.

#### DESCRIPTION

Specifies a brief description for the group. The double quotation marks are required.

#### NEW CONTACTS

Specifies a comma-separated list of contacts that belong to this group. This list overwrites the existing list of contacts for the group.

### CONTACTS ADD

Specifies a comma-separated list of contacts to add to this group.

### CONTACTS 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** *group-name* ►►

### Parameters

#### GROUP

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.

## 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** *SCHEMA cap-schema* **MONITOR-QUALIFIER** *mon-qual* ►►

NOTIFY ( *CONTACT contact-name* | *GROUP group-name* ) ( *STATUS DOWN* | *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 DAS. 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** option, but can be useful if you don't run the DAS 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.

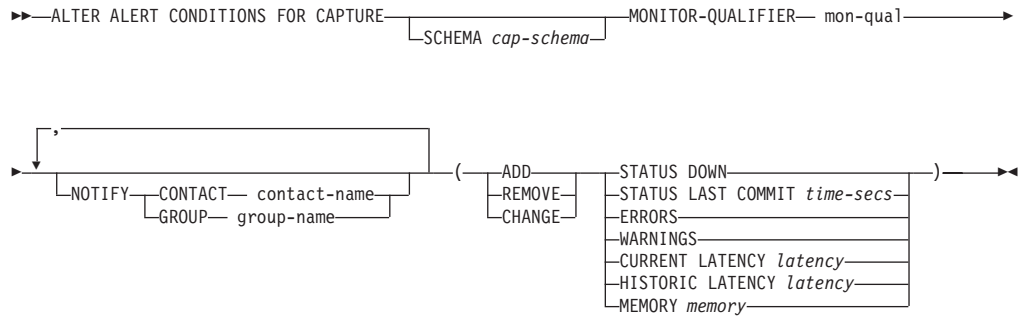
## Usage Notes

- 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.

## ALTER ALERT CONDITIONS FOR CAPTURE command

The ALTER ALERT CONDITIONS FOR CAPTURE command alters alert conditions for the Capture program.

### Syntax



### 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 DAS. 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 option, but can be useful if you don't run the DAS 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—SCHEMA *cap-schema*—MONITOR QUALIFIER *mon-qual*—►►

## Parameters

### SCHEMA

Specifies the Capture schema that qualifies the Capture tables to be monitored. This parameter is required.

### MONITOR QUALIFIER

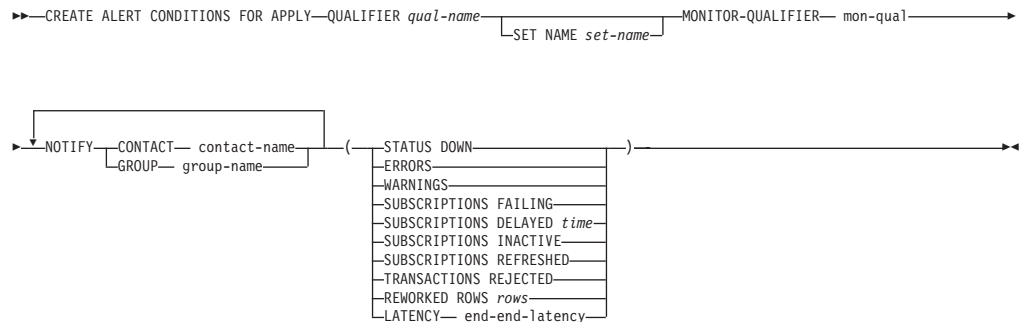
Specifies the Monitor qualifier. This parameter is required.



## 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



### 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.

#### 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 DAS 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 have been 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 have been logged in the IBMSNAP\_APPLYTRACE 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.

### SUBSCRIPTIONS FAILING

Specifies whether the Monitor program checks if processed subscription sets have 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.

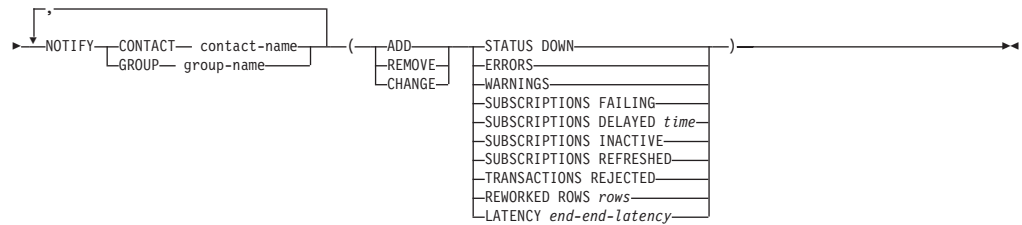
---

## 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—QUALIFIER qual-name [SET NAME set-name] MONITOR-QUALIFIER— mon-qual▶▶
```



## 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 DAS 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 have been 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 have been logged in the IBMSNAP\_APPLYTRACE 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.

### SUBSCRIPTIONS FAILING

Specifies whether the Monitor program checks if processed subscription sets have 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—SCHEMA *cap-schema*—MONITOR 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 9. Example of running the `asnclp` program

The following examples show how the `asnclp` program can be run from the command line so the `SET` and `task` commands work together to set up a replication environment.

---

### Creating databases and tables

The `srcddl.sql` file creates a source and a target database within the current DB2 instance and creates a source table for replication. From a `db2cmd` prompt, run the following command:

```
db2 -vtf srcddl.sql
```

---

### Creating input and output files for the `asnclp` program

Use the following procedures to create input and output files.

#### Input files

The `repl.in` file contains the various `SET` and `task` commands for the `asnclp` program to set up a simple update-anywhere replication environment. From the Windows or UNIX command prompt, or from the `db2cmd` prompt, run the following command:

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

#### Output files

This command produces the following output files in the same directory where you run the `asnclp` program:

- `capture.sql`
- `control.sql`
- `target.sql`
- `replica_ctl_tbls.sql`
- `repl.log`

The `repl.in` file specifies that the `asnclp` program should run the commands immediately, so if you run it, you do not need to run any of the SQL files listed above.



---

## Chapter 10. Frequently Asked Questions

This section contains answers to some of the issues that you may encounter while using the **asnclp** command.

**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, “Running the **asnclp** program”, on page 3.

**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:** What command do I use to create capture control tables on a workstation?

**Response:** Use the following command:

```
create control tables for capture server in uw;
```

**Note:** this command creates tables in the DB2 default tablespace and uses the information you provided in the **SET SERVER** command to find out the server information.

**Question:** If I “set capture schema target xxx”, is it used during create control tables command?

**Response:** This command is used only in auto-registration of a replica or CCD at the target server.

**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:** Does the **asnclp** command require tables and views to exist in the database before executing any replication tasks?

**Response:** Yes, tables and views must already exist in the database before any replication task can be executed. Here are some basic rules to follow:

- The source tables have to exist in the database so that they can be registered
- Source views have to exist in the database so that they can be registered
- Target tables *don't* need to exist in the database; they can be created as part of the subscription member
- Target views *have* to exist in the database to be part of a subscription target
- At least one table in the view definition *must* be registered, the other non-registered tables of the view is treated as static tables.

**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 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 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:** They 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 tablespace 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:** The following are some basic rules:

1. If you do not specify the **IN** clause, then **asnclp** uses the DB2 defaults
2. If you do specify the **IN** clause is specified with a *tsname*:
  - a. If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile
  - b. If you want to reuse an existing table space, include the **REUSE** clause.
  - c. If you do not have a profile, specify the table space name with no profile, and **asnclp** assumes that the table space exists.



3. If you specify **aNAMING PREFIX**, same logic as 2, except that the command line generates the *ts* name
  - a. If you want to create a table space using a profile, include the **CREATE USING PROFILE** clause and specify the name of the profile
  - b. If you want to reuse an existing table space, include the **REUSE** clause.
  - c. If you do not have a profile, specify the table space name with no profile, and **asnclp** assumes that the table space exists.

**NOTE:** This can be an heterogeneous segment or table space name (if so, it needs to already exist).

**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 **BUFFERPOOL** name.

**Question:** How can I use the profile once it has been set?

**Response:** After you invoke the **SET PROFILE** command, you give it a name as follows:

```
SET PROFILE myprof options
```

You can then use the profile during **CREATE REGISTRATION** as follows:

```
CREATE REGISTRATION (ibm.table1) DIFFERENTIAL REFRESH
  STAGE ibm.cd_table1 IN mytablespace
  CREATE USING PROFILE myprof options
```

Once *mytablespace* is created, *mytablespace* will use the properties 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 **asnclp** reaches **CREATE REGISTRATION** command, control tables would already have been created in the database.

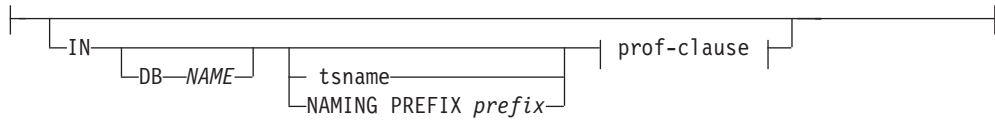
**Question:** Why do I get an SQL0805N message when I connect to a remote z/OS system for the first time?

**Response:** You will get this message if you have not run the **db2 bind** command. See the notes section in Chapter 2, "Running the **asnclp** program", on page 3.

**Question:** How does **NAMING PREFIX** work?

**Response:** Consider the following syntax from **CREATE MEMBER**

**trg-def-clause:**



If the **NAMING PREFIX** keyword is specified, the prefix value provided by the user is used to generate the tablespace name. For example, if the target table name is *MYTGT* and the user specified the following:

```
IN NAMING PREFIX IBM CREATE USING PROFILE MYPROF
```

Then, the tablespace name will be *IBMMYTGT*.

---

# Index

## A

- accessing
  - output files 4
- ACTIVATE
  - ALTER SUBSCRIPTION 33
  - CREATE MEMBER 36
  - CREATE SUBSCRIPTION 31
- ADD
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - ALTER REGISTRATION 26
  - administrative commands
    - command line interface 4
      - running 4
    - input file 4
      - running 4
- AFTER
  - OFFLINE LOAD 47
- ALERTS
  - CREATE CONTROL TABLES 19
- ALL
  - SET SERVER 9
- ALTER ALERT CONDITIONS FOR APPLY
  - ADD 59
  - APPLY QUALIFIER 59
  - CHANGE 59
  - command 5, 58
  - ERRORS 59
  - LATENCY 60
  - MONITOR QUALIFIER 59
  - NOTIFY 59
    - CONTACT 59
    - GROUP 59
  - parameters 59
    - ADD 59
    - APPLY QUALIFIER 59
    - CHANGE 59
    - ERRORS 59
    - LATENCY 60
    - MONITOR QUALIFIER 59
    - NOTIFY 59
    - REMOVE 59
    - REWORKED ROWS 60
    - SET NAME 59
    - STATUS DOWN 59
    - SUBSCRIPTIONS DELAYED 59
    - SUBSCRIPTIONS FAILING 59
    - SUBSCRIPTIONS INACTIVE 60
    - SUBSCRIPTIONS
      - REFRESHED 60
    - TRANSACTIONS REJECTED 60
    - WARNINGS 59
  - REMOVE 59
  - REWORKED ROWS 60
  - SET NAME 59
  - STATUS DOWN 59
  - SUBSCRIPTIONS DELAYED 59
  - SUBSCRIPTIONS FAILING 59
- ALTER ALERT CONDITIONS FOR APPLY (*continued*)
  - SUBSCRIPTIONS INACTIVE 60
  - SUBSCRIPTIONS REFRESHED 60
  - syntax 58
  - TRANSACTIONS REJECTED 60
  - usage notes 60
  - WARNINGS 59
- ALTER ALERT CONDITIONS FOR CAPTURE
  - ADD 55
  - CHANGE 55
  - command 5, 55
  - CURRENT LATENCY 56
  - ERRORS 55
  - HISTORIC LATENCY 56
  - MEMORY 56
  - MONITOR QUALIFIER 55
  - NOTIFY 55
    - CONTACT 55
    - GROUP 55
  - parameters 55
    - ADD 55
    - CHANGE 55
    - CURRENT LATENCY 56
    - ERRORS 55
    - HISTORIC LATENCY 56
    - MEMORY 56
    - MONITOR QUALIFIER 55
    - NOTIFY 55
    - REMOVE 55
    - SCHEMA 55
    - STATUS DOWN 55
    - STATUS LAST COMMIT 55
    - WARNINGS 56
  - REMOVE 55
  - SCHEMA 55
  - STATUS DOWN 55
  - STATUS LAST COMMIT 55
  - syntax 55
  - usage notes 56
  - WARNINGS 56
- ALTER CONTACT
  - command 5, 50
  - CONTACT 50
  - EMAIL 50
  - PAGE 50
  - parameters 50
    - CONTACT 50
    - EMAIL 50
    - PAGE 50
    - SUBSCRIPTION 50
  - SUBSCRIPTION 50
  - syntax 50
- ALTER GROUP
  - command 5, 52
  - CONTACTS ADD 52
  - CONTACTS REMOVE 53
  - DESCRIPTION 52
  - GROUP 52
  - NEW CONTACTS 52
- ALTER GROUP (*continued*)
  - parameters 52
    - CONTACTS ADD 52
    - CONTACTS REMOVE 53
    - DESCRIPTION 52
    - GROUP 52
    - NEW CONTACTS 52
  - syntax 52
- ALTER MEMBER ADD COLS
  - APPLYQUAL 42
  - COLS 42
    - EXPRESSION 42
    - TARGET 42
  - command 5, 42
  - parameters 42
    - APPLYQUAL 42
    - COLS 42
    - SETNAME 42
    - SOURCE 42
    - TARGET 42
  - SETNAME 42
  - SOURCE 42
  - syntax 42
  - TARGET 42
  - usage notes 42
- ALTER REGISTRATION
  - ADD 26
    - COL 26
    - colname 26
    - IMAGE AFTER 26
    - IMAGE BEFORE 26
    - IMAGE BOTH 26
    - PREFIX 26
  - CAPTURE 26
    - ALL 26
    - CHANGES 26
  - command 5, 24
  - FORWARDING 26
  - FULL REFRESH 26
    - parameters 26
      - ADD 26
      - CAPTURE 26
      - FORWARDING 26
      - FULL REFRESH 26
      - ROW 26
      - STOP ON ERROR 26
  - ROW 26
    - CONFLICT 26
    - UPDATE AS DELETE INSERT 26
  - STOP ON ERROR 26
  - syntax 25
  - usage notes 27
- ALTER SUBSCRIPTION
  - ACTIVATE 33
    - NO 33
    - ONCE 33
    - YES 33
  - APPLYQUAL 33
  - BLOCKING 34
  - command 33
  - COMMIT COUNT 34

ALTER SUBSCRIPTION (*continued*)  
 NONIBM SOURCE SERVER 34  
 parameters 33  
   ACTIVATE 33  
   APPLYQUAL 33  
   BLOCKING 34  
   COMMIT COUNT 34  
   NONIBM SOURCE SERVER 34  
   SETNAME 33  
   SETTYPE 33  
   START DATE 34  
   TIME 34  
   TIMING 33  
 SETNAME 33  
 SETTYPE 33  
   F only 33  
   P 33  
   R 33  
   S only 33  
   U 33  
 START DATE 34  
 syntax 33  
 TIME 34  
 TIMING 33  
   BOTH 33  
   CONTINUOUS 34  
   EVENT 33  
   INTERVAL 33  
 ALTER SUBSCRIPTION SET  
 command 5  
 apply control server  
   CREATE CONTROL TABLES 18  
   DROP CONTROL TABLES 20  
 Apply control server  
   SET SERVER 7  
 APPLY QUALIFIER  
   ALTER ALERT CONDITIONS FOR  
     APPLY 59  
   CREATE ALERT CONDITIONS FOR  
     APPLY 57  
   DROP ALERT CONDITIONS FOR  
     APPLY 60  
 APPLYQUAL  
   ALTER MEMBER ADD COLS 42  
   ALTER SUBSCRIPTION 33  
   CREATE MEMBER 36  
   CREATE STMT 43  
   CREATE SUBSCRIPTION 31  
   DROP MEMBER 41  
   DROP STMT 44  
   DROP SUBSCRIPTION SET 44  
   OFFLINE LOAD 47  
   PROMOTE SUBSCRIPTION SET 45,  
     46  
 ARCHLEVEL  
   DROP CONTROL TABLES 20  
 AS400 HOSTNAME  
   SET SERVER 9  
 asncp  
 compare 1  
 asncp program  
 running  
   example 61

## B

BEFORE  
 OFFLINE LOAD 47  
 BLOCKING  
   ALTER SUBSCRIPTION 34

## C

CAPTURE  
   ALTER REGISTRATION 26  
   SET SERVER 9  
 Capture control server  
   SET SERVER 7  
 CAPTURE SCHEMA  
   PROMOTE REGISTRATION 28  
 capture schema information  
   SET CAPTURE SCHEMA 8  
 CAPTURE SCRIPT  
   SET LOG 14  
   SET OUTPUT 14  
 capture server  
   CREATE CONTROL TABLES 18  
   DROP CONTROL TABLES 20  
 CCD  
   CREATE MEMBER 40  
 CD SCHEMA  
   PROMOTE REGISTRATION 28  
 CHANGE  
   ALTER ALERT CONDITIONS FOR  
     APPLY 59  
   ALTER ALERT CONDITIONS FOR  
     CAPTURE 55  
 COL  
   CREATE REGISTRATION 23  
 COLS  
   ALTER MEMBER ADD COLS 42  
   CREATE MEMBER 40  
 command  
   ALTER ALERT CONDITIONS FOR  
     APPLY 5, 58  
   ALTER ALERT CONDITIONS FOR  
     CAPTURE 5, 55  
   ALTER CONTACT 5, 50  
   ALTER GROUP 5, 52  
   ALTER MEMBER ADD COLS 5, 42  
   ALTER REGISTRATION 5, 24  
   ALTER SUBSCRIPTION 33  
   ALTER SUBSCRIPTION SET 5  
   CREATE ALERT CONDITIONS FOR  
     APPLY 5, 57  
   CREATE ALERT CONDITIONS FOR  
     CAPTURE 5, 53  
   CREATE CONTACT 5, 49  
   CREATE CONTROL TABLES 5, 17  
   CREATE GROUP 5, 52  
   CREATE MEMBER 5, 34  
   CREATE REGISTRATION 5, 21  
   CREATE STMT 5, 42  
   CREATE SUBSCRIPTION 31  
   CREATE SUBSCRIPTION SET 5  
   DELEGATE CONTACT 5, 51  
   DROP ALERT CONDITIONS FOR  
     APPLY 5, 60  
   DROP ALERT CONDITIONS FOR  
     CAPTURE 5, 56  
   DROP CONTACT 5, 50

command (*continued*)  
   DROP CONTROL TABLES 5, 19  
   DROP GROUP 5, 53  
   DROP MEMBER 5, 41  
   DROP REGISTRATION 5, 27  
   DROP STMT 5, 44  
   DROP SUBSCRIPTION SET 5, 44  
   OFFLINE LOAD 5, 47  
   PROMOTE REGISTRATION 5, 27  
   PROMOTE SUBSCRIPTION SET 5,  
     45  
   SET CAPTURE SCHEMA 5, 14  
   SET DROP 5, 13  
   SET LOG 14  
   SET OUTPUT 14  
   SET OUTPUT, SET LOG 5  
   SET PROFILE 5, 10  
   SET RUN SCRIPT 5, 15  
   SET SERVER 5, 8  
   SET TRACE 5, 15  
   SUBSTITUTE CONTACT 5, 51  
 command line interface  
   administrative commands 4  
 commands  
   control table definition 17  
   Monitor table definition 49  
   offline load utility 47  
   registration definition 21  
   SET 7  
   setup 5  
   subscription definition 31  
   task 5  
 COMMIT COUNT  
   ALTER SUBSCRIPTION 34  
 compare  
   asncp 1  
 COMPLETE  
   CREATE MEMBER 40  
 CONDENSED  
   CREATE MEMBER 40  
   CREATE REGISTRATION 22  
 CONTACT  
   ALTER CONTACT 50  
   CREATE CONTACT 49, 50  
   DELEGATE CONTACT 51  
   DROP CONTACT 50  
   SUBSTITUTE CONTACT 51  
 CONTACTS  
   CREATE GROUP 52  
 CONTACTS ADD  
   ALTER GROUP 52  
 CONTACTS REMOVE  
   ALTER GROUP 53  
 CONTROL  
   SET SERVER 9  
 CONTROL SCRIPT  
   SET LOG 14  
   SET OUTPUT 14  
 control table commands  
   SET SERVER 8  
 control table definition  
   commands 17  
 CREATE ALERT CONDITIONS FOR  
   APPLY  
     APPLY QUALIFIER 57  
     command 5, 57  
   ERRORS 57

CREATE ALERT CONDITIONS FOR  
 APPLY (continued)  
 LATENCY 58  
 MONITOR QUALIFIER 57  
 NOTIFY 57  
   CONTACT 57  
   GROUP 57  
 parameters 57  
   APPLY QUALIFIER 57  
   ERRORS 57  
   LATENCY 58  
   MONITOR QUALIFIER 57  
   NOTIFY 57  
   REWORKED ROWS 58  
   SET NAME 57  
   STATUS DOWN 57  
   SUBSCRIPTIONS DELAYED 58  
   SUBSCRIPTIONS FAILING 57  
   SUBSCRIPTIONS INACTIVE 58  
   SUBSCRIPTIONS  
     REFRESHED 58  
   TRANSACTIONS REJECTED 58  
   WARNINGS 57  
 REWORKED ROWS 58  
 SET NAME 57  
 STATUS DOWN 57  
 SUBSCRIPTIONS DELAYED 58  
 SUBSCRIPTIONS FAILING 57  
 SUBSCRIPTIONS INACTIVE 58  
 SUBSCRIPTIONS REFRESHED 58  
 syntax 57  
 TRANSACTIONS REJECTED 58  
 usage notes 58  
 WARNINGS 57  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE  
 command 5, 53  
 CURRENT LATENCY 54  
 ERRORS 54  
 HISTORIC LATENCY 54  
 MEMORY 54  
 MONITOR QUALIFIER 53  
 NOTIFY 53  
   CONTACT 54  
   GROUP 54  
 parameters 53  
   CURRENT LATENCY 54  
   ERRORS 54  
   HISTORIC LATENCY 54  
   MEMORY 54  
   MONITOR QUALIFIER 53  
   NOTIFY 53  
   SCHEMA 53  
   STATUS DOWN 54  
   STATUS LAST COMMIT 54  
   WARNINGS 54  
 SCHEMA 53  
 STATUS DOWN 54  
 STATUS LAST COMMIT 54  
 syntax 53  
 usage notes 54  
 WARNINGS 54  
 CREATE CONTACT  
 command 5, 49  
 CONTACT 49, 50  
 EMAIL 49  
 GROUP 49

CREATE CONTACT (continued)  
 parameters 49  
   CONTACT 49, 50  
   EMAIL 49  
   GROUP 49  
   SUBSCRIPTION 50  
 SUBSCRIPTION 50  
 syntax 49  
 CREATE CONTROL TABLES  
 ALERTS 19  
 apply control server 18  
 capture server 18  
 command 5, 17  
 CREATE USING PROFILE  
 clause 18  
 DB 19  
 IN 18  
 monitor control server 18  
 OTHERS 19  
 PAGE LOCK 19  
 parameters 18  
   ALERTS 19  
   apply control server 18  
   capture server 18  
   DB 19  
   IN 18  
   monitor control server 18  
   OTHERS 19  
   PAGE LOCK 19  
   ROW LOCK 19  
   SCHEMA 19  
   UOW 19  
 REUSE  
 clause 18  
 ROW LOCK 19  
 SCHEMA 19  
 syntax 17  
 UOW 19  
 usage notes 19  
 CREATE GROUP  
 command 5, 52  
 CONTACTS 52  
 DESCRIPTION 52  
 GROUP 52  
 parameters 52  
   CONTACTS 52  
   DESCRIPTION 52  
   GROUP 52  
 syntax 52  
 CREATE MEMBER  
 ACTIVATE 36  
   NO 36  
   ONCE 36  
   YES 36  
 APPLYQUAL 36  
 CCD 40  
   AS SOURCE 40  
   WITH UOW COLS 40  
 COLS 40  
   ALL REGISTERED 40  
   INCLUDE 40  
 command 5, 34  
 COMPLETE 40  
   OFF 40  
   ON 40  
 CONDENSED 40  
   OFF 40

CREATE MEMBER (continued)  
 CONDENSED (continued)  
   ON 40  
 CREATE USING PROFILE  
 clause 37  
 EXCLUDE 41  
 EXTERNAL 40  
 FORWARDING 39  
   OFF 39  
   ON 39  
 FULL REFRESH 39  
   OFF 39  
   ON 39  
 IN 37  
 INTERNAL 40  
 KEYS 41  
 LOADX TYPE 36  
   CROSSLOADER 37  
   IMPORT EXPORT 37  
   LOAD EXPORT 37  
   NO ASNLOAD 36  
   USER DEFINED 36  
 parameters 36  
   ACTIVATE 36  
   APPLYQUAL 36  
   CCD 40  
   COLS 40  
   COMPLETE 40  
   CONDENSED 40  
   EXCLUDE 41  
   EXTERNAL 40  
   FORWARDING 39  
   FULL REFRESH 39  
   IN 37  
   INTERNAL 40  
   KEYS 41  
   LOADX TYPE 36  
   SETNAME 36  
   SOURCE 37  
   STOP ON ERROR 39  
   TARGET 37  
   TGT KEY CHANGE 40  
   TYPE 38  
   UPDATE AS DELETE INSERT 39  
   WHERE 40  
 REUSE  
 clause 37  
 SETNAME 36  
 SOURCE 37  
 STOP ON ERROR 39  
   OFF 40  
   ON 39  
 syntax 34  
 TARGET 37  
   DEFINITION 37  
   NAME 37  
   NAMING PREFIX 37  
 TGT KEY CHANGE 40  
   OFF 40  
   ON 40  
 TYPE 38  
   BASEAGGREGATE 38  
   CHANGEAGGREGATE 38  
   PIT 38  
   REPLICA 38  
   USERCOPY 38  
 UPDATE AS DELETE INSERT 39

CREATE MEMBER (continued)  
 OFF 39  
 ON 39  
 usage notes 41  
 WHERE 40  
 CREATE REGISTRATION  
 COL 23  
 ALL 23  
 BOTH 23  
 colname 23  
 IMAGE AFTER 23  
 PREFIX 23  
 command 5, 21  
 CONDENSED 22  
 OFF 23  
 ON 23  
 CREATE USING PROFILE  
 clause 23  
 DIFFERENTIAL REFRESH 22  
 FULL REFRESH ONLY 23  
 IN 23  
 NONIBM 23  
 remote ccd NAME 23  
 remote ccd owner 23  
 objname 22  
 objowner 22  
 OPTIONS 24  
 CONFLICT 24  
 UPDATE AS DELETE INSERT 24  
 parameters 22  
 CAPTURE 24  
 COL 23  
 CONDENSED 22  
 DIFFERENTIAL REFRESH 22  
 FORWARDING 24  
 FULL REFRESH 24  
 FULL REFRESH ONLY 23  
 IN 23  
 NONIBM 23  
 objname 22  
 objowner 22  
 OPTIONS 24  
 RMTJRN 22  
 STAGE 22  
 STOP ON ERROR 24  
 REUSE  
 clause 23  
 RMTJRN 22  
 STAGE 22  
 syntax 21  
 usage notes 24  
 CREATE SOURCE WITH SCHEMA  
 PROMOTE REGISTRATION 28  
 CREATE STMT  
 APPLYQUAL 43  
 command 5, 42  
 EXECUTE 43  
 AFTER AT TARGET 43  
 AT SOURCE 43  
 BEFORE AT TARGET 43  
 SQLSTATES 43  
 NUMBER 43  
 parameters 43  
 APPLYQUAL 43  
 EXECUTE 43  
 NUMBER 43  
 PROC 43

CREATE STMT (continued)  
 parameters (continued)  
 SETNAME 43  
 SETTYPE 43  
 SQL 43  
 PROC 43  
 SETNAME 43  
 SETTYPE 43  
 P 43  
 R 43  
 U 43  
 SQL 43  
 syntax 43  
 CREATE SUBSCRIPTION  
 ACTIVATE 31  
 NO 31  
 ONCE 31  
 YES 31  
 APPLYQUAL 31  
 BLOCKING 32  
 command 31  
 COMMIT COUNT 32  
 NONIBM SOURCE SERVER 32  
 parameters 31  
 ACTIVATE 31  
 APPLYQUAL 31  
 BLOCKING 32  
 COMMIT COUNT 32  
 NONIBM SOURCE SERVER 32  
 SETNAME 31  
 SETTYPE 32  
 START DATE 32  
 TIME 32  
 TIMING 32  
 SETNAME 31  
 SETTYPE 32  
 P 32  
 R 32  
 U 32  
 START DATE 32  
 syntax 31  
 TIME 32  
 TIMING 32  
 BOTH 32  
 CONTINUOUS 32  
 EVENT 32  
 INTERVAL 32  
 usage notes 32  
 CREATE SUBSCRIPTION SET  
 command 5  
 CREATE USING PROFILE  
 clause  
 CREATE CONTROL TABLES 18  
 CREATE MEMBER 37  
 CREATE REGISTRATION 23  
 creating  
 example databases 61  
 example tables 61  
 input files 61  
 output files 61  
 CURRENT LATENCY  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 56  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 54  
 customization  
 SET PROFILE 7

**D**  
 DB  
 CREATE CONTROL TABLES 19  
 SET SERVER 9  
 DBALIAS  
 SET SERVER 9  
 DBNAME  
 SET SERVER 9  
 DEFAULT  
 SET CAPTURE SCHEMA 15  
 DELEGATE CONTACT  
 command 5, 51  
 CONTACT 51  
 FROM 51  
 parameters 51  
 CONTACT 51  
 FROM 51  
 TO 51  
 syntax 51  
 TO 51  
 DESCRIPTION  
 ALTER GROUP 52  
 CREATE GROUP 52  
 DIFFERENTIAL REFRESH  
 CREATE REGISTRATION 22  
 DROP ALERT CONDITIONS FOR  
 APPLY  
 APPLY QUALIFIER 60  
 command 5, 60  
 MONITOR QUALIFIER 60  
 parameters 60  
 APPLY QUALIFIER 60  
 MONITOR QUALIFIER 60  
 syntax 60  
 DROP ALERT CONDITIONS FOR  
 CAPTURE  
 command 5, 56  
 MONITOR QUALIFIER 56  
 parameters 56  
 MONITOR QUALIFIER 56  
 SCHEMA 56  
 SCHEMA 56  
 syntax 56  
 DROP CONTACT  
 command 5, 50  
 CONTACT 50  
 parameters 50  
 CONTACT 50  
 SUBSTITUTE WITH 50  
 SUBSTITUTE WITH 50  
 syntax 50  
 usage notes 51  
 DROP CONTROL TABLES  
 apply control server 20  
 ARCHLEVEL 20  
 capture server 20  
 command 5, 19  
 monitor control server 20  
 NONIBM SCHEMA 20  
 parameters 20  
 apply control server 20  
 ARCHLEVEL 20  
 capture server 20  
 monitor control server 20  
 NONIBM SCHEMA 20  
 syntax 20  
 usage notes 20

DROP GROUP  
 command 5, 53  
 GROUP 53  
 parameters 53  
 GROUP 53  
 syntax 53  
 usage notes 53

drop information  
 SET DROP 8

DROP MEMBER  
 APPLYQUAL 41  
 command 5, 41  
 parameters 41  
 APPLYQUAL 41  
 SETNAME 41  
 SOURCE 41  
 TARGET 41  
 SETNAME 41  
 SOURCE 41  
 syntax 41  
 TARGET 41  
 usage notes 41

DROP REGISTRATION  
 command 5, 27  
 objname 27  
 objowner 27  
 parameters 27  
 objname 27  
 objowner 27  
 syntax 27  
 usage notes 27

DROP STMT  
 APPLYQUAL 44  
 command 5, 44  
 NUMBER 44  
 parameters 44  
 APPLYQUAL 44  
 NUMBER 44  
 SETNAME 44  
 SETTYPE 44  
 SETNAME 44  
 SETTYPE 44  
 P 44  
 R 44  
 U 44  
 syntax 44  
 usage notes 44

DROP SUBSCRIPTION SET  
 APPLYQUAL 44  
 command 5, 44  
 parameters 44  
 APPLYQUAL 44  
 SETNAME 44  
 SETNAME 44  
 syntax 44  
 usage notes 45

## E

EMAIL  
 ALTER CONTACT 50  
 CREATE CONTACT 49

environmental  
 setup  
 commands 5

ERRORS  
 ALTER ALERT CONDITIONS FOR  
 APPLY 59  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 55  
 CREATE ALERT CONDITIONS FOR  
 APPLY 57  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 54

example  
 running  
 asncpl program 61

example databases  
 creating 61

example tables  
 creating 61

EXCLUDE  
 CREATE MEMBER 41

EXECUTE  
 CREATE STMT 43

EXTERNAL  
 CREATE MEMBER 40

## F

FOR OBJECT  
 SET PROFILE 11

FORWARDING  
 ALTER REGISTRATION 26  
 CREATE MEMBER 39

frequently asked questions 63

FROM  
 DELEGATE CONTACT 51

FULL REFRESH  
 ALTER REGISTRATION 26  
 CREATE MEMBER 39

FULL REFRESH ONLY  
 CREATE REGISTRATION 23

## G

GROUP  
 ALTER GROUP 52  
 CREATE CONTACT 49  
 CREATE GROUP 52  
 DROP GROUP 53

## H

HISTORIC LATENCY  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 56  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 54

## I

ID  
 SET SERVER 10

IN  
 CREATE CONTROL TABLES 18  
 CREATE MEMBER 37  
 CREATE REGISTRATION 23

input file  
 administrative commands 4

input files  
 creating 61

INTERNAL  
 CREATE MEMBER 40

## J

Java environment  
 setting up 3

## K

KEYS  
 CREATE MEMBER 41

## L

LATENCY  
 ALTER ALERT CONDITIONS FOR  
 APPLY 60  
 CREATE ALERT CONDITIONS FOR  
 APPLY 58

LATER  
 SET RUN SCRIPT 16

LOADX TYPE  
 CREATE MEMBER 36

LOG NAME  
 SET LOG 14  
 SET OUTPUT 14

## M

MEMORY  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 56  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 54

MONITOR  
 SET SERVER 9

Monitor administration commands  
 SET SERVER 9

monitor control server  
 CREATE CONTROL TABLES 18  
 DROP CONTROL TABLES 20

Monitor control server  
 SET SERVER 7

MONITOR QUALIFIER  
 ALTER ALERT CONDITIONS FOR  
 APPLY 59  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 55  
 CREATE ALERT CONDITIONS FOR  
 APPLY 57  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 53  
 DROP ALERT CONDITIONS FOR  
 APPLY 60  
 DROP ALERT CONDITIONS FOR  
 CAPTURE 56

MONITOR SCRIPT  
 SET LOG 14  
 SET OUTPUT 14

Monitor table definition  
 commands 49

## N

- NEW CONTACTS
  - ALTER GROUP 52
- NONIBM
  - CREATE REGISTRATION 23
- NONIBM SCHEMA
  - DROP CONTROL TABLES 20
- NONIBM SERVER
  - SET SERVER 9
- NONIBM SOURCE SERVER
  - ALTER SUBSCRIPTION 34
- NOTIFY
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - CREATE ALERT CONDITIONS FOR APPLY 57
  - CREATE ALERT CONDITIONS FOR CAPTURE 53
- NOW
  - SET RUN SCRIPT 16
- NULLS
  - SET CAPTURE SCHEMA 15
  - SET SERVER 9
- NUMBER
  - CREATE STMT 43
  - DROP STMT 44

## O

- objname
  - CREATE REGISTRATION 22
  - DROP REGISTRATION 27
  - PROMOTE REGISTRATION 28
- objowner
  - CREATE REGISTRATION 22
  - DROP REGISTRATION 27
  - PROMOTE REGISTRATION 28
- OFF
  - SET TRACE 15
- OFFLINE LOAD
  - AFTER 47
  - APPLYQUAL 47
  - BEFORE 47
  - command 5, 47
  - parameters 47
    - AFTER 47
    - APPLYQUAL 47
    - BEFORE 47
    - SETNAME 47
  - SETNAME 47
  - syntax 47
- offline load utility
  - commands 47
- ON
  - SET TRACE 15
- Operating systems
  - supported 2
- OPTIONS
  - CREATE REGISTRATION 24
- OTHERS
  - CREATE CONTROL TABLES 19
- output files
  - accessing 4
  - creating 61

- output information
  - SET LOG 8
  - SET OUTPUT 8

## P

- PAGE
  - ALTER CONTACT 50
- PAGE LOCK
  - CREATE CONTROL TABLES 19
- parameters
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - ALTER CONTACT 50
  - ALTER GROUP 52
  - ALTER MEMBER ADD COLS 42
  - ALTER REGISTRATION 26
  - ALTER SUBSCRIPTION 33
  - CREATE ALERT CONDITIONS FOR APPLY 57
  - CREATE ALERT CONDITIONS FOR CAPTURE 53
  - CREATE CONTACT 49
  - CREATE CONTROL TABLES 18
  - CREATE GROUP 52
  - CREATE MEMBER 36
  - CREATE REGISTRATION 22
  - CREATE STMT 43
  - CREATE SUBSCRIPTION 31
  - DELEGATE CONTACT 51
  - DROP ALERT CONDITIONS FOR APPLY 60
  - DROP ALERT CONDITIONS FOR CAPTURE 56
  - DROP CONTACT 50
  - DROP CONTROL TABLES 20
  - DROP GROUP 53
  - DROP MEMBER 41
  - DROP REGISTRATION 27
  - DROP STMT 44
  - DROP SUBSCRIPTION SET 44
  - OFFLINE LOAD 47
  - PROMOTE REGISTRATION 28
  - PROMOTE SUBSCRIPTION SET 45
  - SET CAPTURE SCHEMA 15
  - SET DROP 13
  - SET LOG 14
  - SET OUTPUT 14
  - SET PROFILE 11
  - SET RUN SCRIPT 16
  - SET SERVER 9
  - SET TRACE 15
  - SUBSTITUTE CONTACT 51
- PASSWORD
  - SET SERVER 10
- PROC
  - CREATE STMT 43
- PROFILE
  - SET PROFILE 11
- PROMOTE REGISTRATION
  - CAPTURE SCHEMA 28
  - CD SCHEMA 28
  - command 5, 27
  - CREATE SOURCE WITH SCHEMA 28

PROMOTE REGISTRATION (*continued*)

- objname 28
- objowner 28
- parameters 28
  - CAPTURE SCHEMA 28
  - CD SCHEMA 28
  - CREATE SOURCE WITH SCHEMA 28
  - objname 28
  - objowner 28
  - SOURCE DB 28
  - SOURCE TABLE 28
  - USING SCHEMA 28
  - VIEW 28
  - WITH UNREGISTERED BASE TABLES 28
- SOURCE DB 28
- SOURCE TABLE 28
- syntax 27
- usage notes 28
- USING SCHEMA 28
- VIEW 28
- WITH UNREGISTERED BASE TABLES 28

PROMOTE SUBSCRIPTION SET

- APPLYQUAL 45, 46
- command 5, 45
- parameters 45
  - APPLYQUAL 45, 46
  - SETNAME 45, 46
  - SOURCE SCHEMA 46
  - TARGET 46
  - USING 45
- SETNAME 45, 46
- SOURCE SCHEMA 46
- syntax 45
- TARGET 46
  - CD SCHEMA 46
  - SCHEMA 46
- usage notes 46
- USING 45
  - CAPTURE SCHEMA FOR 45
  - DB FOR 45

## R

- registration commands
  - SET SERVER 8
- registration definition
  - commands 21
- REMOTE SOURCE
  - SET SERVER 9
- REMOVE
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
- REUSE
  - clause
    - CREATE CONTROL TABLES 18
    - CREATE MEMBER 37
    - CREATE REGISTRATION 23
- REWORKED ROWS
  - ALTER ALERT CONDITIONS FOR APPLY 60
  - CREATE ALERT CONDITIONS FOR APPLY 58



RMTJRN  
 CREATE REGISTRATION 22  
 ROW  
 ALTER REGISTRATION 26  
 ROW LOCK  
 CREATE CONTROL TABLES 19  
 run option  
 SET RUN SCRIPT 8  
 running  
 example  
 asncp program 61

**S**

SCHEMA  
 ALTER ALERT CONDITIONS FOR  
 CAPTURE 55  
 CREATE ALERT CONDITIONS FOR  
 CAPTURE 53  
 CREATE CONTROL TABLES 19  
 DROP ALERT CONDITIONS FOR  
 CAPTURE 56

SET  
 commands 7

SET CAPTURE SCHEMA  
 capture schema information 8  
 command 5, 14  
 DEFAULT 15  
 NULLS 15  
 parameters 15  
 DEFAULT 15  
 NULLS 15  
 SOURCE 15  
 TARGET 15  
 SOURCE 15  
 syntax 14  
 TARGET 15

SET DROP  
 command 5, 13  
 drop information 8  
 parameters 13  
 TARGET 13  
 syntax 13  
 TARGET 13  
 usage notes 13

SET LOG  
 CAPTURE SCRIPT 14  
 command 14  
 CONTROL SCRIPT 14  
 LOG NAME 14  
 MONITOR SCRIPT 14  
 output information 8  
 parameters 14  
 CAPTURE SCRIPT 14  
 CONTROL SCRIPT 14  
 LOG NAME 14  
 MONITOR SCRIPT 14  
 TARGET SCRIPT 14  
 syntax 14  
 TARGET SCRIPT 14  
 usage notes 14

SET NAME  
 ALTER ALERT CONDITIONS FOR  
 APPLY 59  
 CREATE ALERT CONDITIONS FOR  
 APPLY 57

SET OUTPUT  
 CAPTURE SCRIPT 14  
 command 14  
 CONTROL SCRIPT 14  
 LOG NAME 14  
 MONITOR SCRIPT 14  
 output information 8  
 parameters 14  
 CAPTURE SCRIPT 14  
 CONTROL SCRIPT 14  
 LOG NAME 14  
 MONITOR SCRIPT 14  
 TARGET SCRIPT 14  
 syntax 14  
 TARGET SCRIPT 14  
 usage notes 14

SET OUTPUT, SET LOG  
 command 5

SET PROFILE  
 capture server 11  
 command 5, 10  
 customization 7  
 FOR OBJECT 11  
 CCD 11  
 CD 11  
 OTHERS 11  
 PAGE LOCK 11  
 ROW LOCK 11  
 TARGET 11  
 UOW 11  
 parameters 11  
 ABSOLUTE 12  
 FOR OBJECT 11  
 PAGESIZE 12  
 PERCENT OF SOURCE 12  
 PRIQTY 12  
 PROFILE 11  
 SECQTY 12  
 STOGROUP 12  
 UNDO 13  
 USING 12  
 syntax 10

TABLESPACE OPTIONS 12  
 BUFFER POOLS 12  
 DB 12  
 ENCODING 12  
 UNDO 13  
 USING  
 SIZE 12

SET RUN SCRIPT  
 command 5, 15  
 LATER 16  
 NOW 16  
 parameters 16  
 LATER 16  
 NOW 16  
 STOP ON SQL ERROR 16  
 run option 8  
 STOP ON SQL ERROR 16  
 OFF 16  
 ON 16  
 syntax 16  
 usage notes 16

SET SERVER  
 ALL 9  
 Apply control server 7  
 AS400 HOSTNAME 9

SET SERVER (continued)  
 CAPTURE 9  
 Capture control server 7  
 command 5, 8  
 CONTROL 9  
 control table commands 8  
 DB 9  
 DBALIAS 9  
 DBNAME 9  
 ID 10  
 MONITOR 9  
 Monitor administration commands 9  
 Monitor control server 7  
 NONIBM SERVER 9  
 NULLS 9  
 parameters 9  
 ALL 9  
 AS400 HOSTNAME 9  
 CAPTURE 9  
 CONTROL 9  
 DB 9  
 DBALIAS 9  
 DBNAME 9  
 ID 10  
 MONITOR 9  
 NONIBM SERVER 9  
 NULLS 9  
 PASSWORD 10  
 REMOTE SOURCE 9  
 TARGET 9  
 PASSWORD 10  
 registration commands 8  
 REMOTE SOURCE 9  
 subscription commands 9  
 syntax 9  
 TARGET 9  
 target server 7  
 usage notes 10

SET TRACE  
 command 5, 15  
 OFF 15  
 ON 15  
 parameters 15  
 OFF 15  
 ON 15  
 syntax 15  
 trace information 8

SETNAME  
 ALTER MEMBER ADD COLS 42  
 ALTER SUBSCRIPTION 33  
 CREATE MEMBER 36  
 CREATE STMT 43  
 CREATE SUBSCRIPTION 31  
 DROP MEMBER 41  
 DROP STMT 44  
 DROP SUBSCRIPTION SET 44  
 OFFLINE LOAD 47  
 PROMOTE SUBSCRIPTION SET 45,  
 46  
 setting up  
 Java environment 3

SETTYPE  
 ALTER SUBSCRIPTION 33  
 CREATE STMT 43  
 CREATE SUBSCRIPTION 32  
 DROP STMT 44

- setup
  - commands 5
- SOURCE
  - ALTER MEMBER ADD COLS 42
  - CREATE MEMBER 37
  - DROP MEMBER 41
  - SET CAPTURE SCHEMA 15
- SOURCE DB
  - PROMOTE REGISTRATION 28
- SOURCE SCHEMA
  - PROMOTE SUBSCRIPTION SET 46
- SOURCE TABLE
  - PROMOTE REGISTRATION 28
- SQL
  - CREATE STMT 43
- STAGE
  - CREATE REGISTRATION 22
- START DATE
  - ALTER SUBSCRIPTION 34
  - CREATE SUBSCRIPTION 32
- START TIME
  - BLOCKING 32
  - COMMIT COUNT 32
  - NONIBM SOURCE SERVER 32
- STATUS DOWN
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - CREATE ALERT CONDITIONS FOR APPLY 57
  - CREATE ALERT CONDITIONS FOR CAPTURE 54
- STATUS LAST COMMIT
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - CREATE ALERT CONDITIONS FOR CAPTURE 54
- STOP ON ERROR
  - ALTER REGISTRATION 26
  - CREATE MEMBER 39
- STOP ON SQL ERROR
  - SET RUN SCRIPT 16
- SUBSCRIPTION
  - ALTER CONTACT 50
  - CREATE CONTACT 50
- subscription commands
  - SET SERVER 9
- subscription definition
  - commands 31
- SUBSCRIPTIONS DELAYED
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - CREATE ALERT CONDITIONS FOR APPLY 58
- SUBSCRIPTIONS FAILING
  - ALTER ALERT CONDITIONS FOR APPLY 59
  - CREATE ALERT CONDITIONS FOR APPLY 57
- SUBSCRIPTIONS INACTIVE
  - ALTER ALERT CONDITIONS FOR APPLY 60
  - CREATE ALERT CONDITIONS FOR APPLY 58
- SUBSCRIPTIONS REFRESHED
  - ALTER ALERT CONDITIONS FOR APPLY 60
  - CREATE ALERT CONDITIONS FOR APPLY 58
- SUBSTITUTE CONTACT
  - command 5, 51
  - CONTACT 51
  - parameters 51
  - CONTACT 51
  - WITH 51
  - syntax 51
  - WITH 51
- SUBSTITUTE WITH
  - DROP CONTACT 50
- supported
  - Operating systems 2
- syntax
  - ALTER ALERT CONDITIONS FOR APPLY 58
  - ALTER ALERT CONDITIONS FOR CAPTURE 55
  - ALTER CONTACT 50
  - ALTER GROUP 52
  - ALTER MEMBER ADD COLS 42
  - ALTER REGISTRATION 25
  - ALTER SUBSCRIPTION 33
  - CREATE ALERT CONDITIONS FOR APPLY 57
  - CREATE ALERT CONDITIONS FOR CAPTURE 53
  - CREATE CONTACT 49
  - CREATE CONTROL TABLES 17
  - CREATE GROUP 52
  - CREATE MEMBER 34
  - CREATE REGISTRATION 21
  - CREATE STMT 43
  - CREATE SUBSCRIPTION 31
  - DELEGATE CONTACT 51
  - DROP ALERT CONDITIONS FOR APPLY 60
  - DROP ALERT CONDITIONS FOR CAPTURE 56
  - DROP CONTACT 50
  - DROP CONTROL TABLES 20
  - DROP GROUP 53
  - DROP MEMBER 41
  - DROP REGISTRATION 27
  - DROP STMT 44
  - DROP SUBSCRIPTION SET 44
  - OFFLINE LOAD 47
  - PROMOTE REGISTRATION 27
  - PROMOTE SUBSCRIPTION SET 45
  - SET CAPTURE SCHEMA 14
  - SET DROP 13
  - SET LOG 14
  - SET OUTPUT 14
  - SET PROFILE 10
  - SET RUN SCRIPT 16
  - SET SERVER 9
  - SET TRACE 15
  - SUBSTITUTE CONTACT 51
- TARGET
  - ALTER MEMBER ADD COLS 42
  - CREATE MEMBER 37
  - DROP MEMBER 41
  - PROMOTE SUBSCRIPTION SET 46
  - SET CAPTURE SCHEMA 15
  - SET SERVER 9
- TARGET SCRIPT
  - SET LOG 14
  - SET OUTPUT 14
- target server
  - SET SERVER 7
- task
  - commands 5
    - control table definitions 17
    - Monitor table definitions 49
    - offline load utility 47
    - registration definitions 21
    - subscription definitions 31
- TGT KEY CHANGE
  - CREATE MEMBER 40
- TIME
  - ALTER SUBSCRIPTION 34
  - CREATE SUBSCRIPTION 32
- TIMING
  - ALTER SUBSCRIPTION 33
  - CREATE SUBSCRIPTION 32
- TO
  - DELEGATE CONTACT 51
- trace information
  - SET TRACE 8
- TRANSACTIONS REJECTED
  - ALTER ALERT CONDITIONS FOR APPLY 60
  - CREATE ALERT CONDITIONS FOR APPLY 58
- TYPE
  - CREATE MEMBER 38

**U**

- UOW
  - CREATE CONTROL TABLES 19
- UPDATE AS DELETE INSERT
  - CREATE MEMBER 39
- usage notes
  - ALTER ALERT CONDITIONS FOR APPLY 60
  - ALTER ALERT CONDITIONS FOR CAPTURE 56
  - ALTER MEMBER ADD COLS 42
  - ALTER REGISTRATION 27
  - CREATE ALERT CONDITIONS FOR APPLY 58
  - CREATE ALERT CONDITIONS FOR CAPTURE 54
  - CREATE CONTROL TABLES 19
  - CREATE MEMBER 41
  - CREATE REGISTRATION 24
  - CREATE SUBSCRIPTION 32
  - DROP CONTACT 51
  - DROP CONTROL TABLES 20
  - DROP GROUP 53
  - DROP MEMBER 41
  - DROP REGISTRATION 27
  - DROP STMT 44
  - DROP SUBSCRIPTION SET 45

**T**

- TABLESPACE OPTIONS
  - SET PROFILE 12

usage notes (*continued*)

- PROMOTE REGISTRATION 28
- PROMOTE SUBSCRIPTION SET 46
- SET DROP 13
- SET LOG 14
- SET OUTPUT 14
- SET RUN SCRIPT 16
- SET SERVER 10

USING

- PROMOTE SUBSCRIPTION SET 45

USING SCHEMA

- PROMOTE REGISTRATION 28

## V

VIEW

- PROMOTE REGISTRATION 28

## W

WARNINGS

- ALTER ALERT CONDITIONS FOR  
APPLY 59
- ALTER ALERT CONDITIONS FOR  
CAPTURE 56
- CREATE ALERT CONDITIONS FOR  
APPLY 57
- CREATE ALERT CONDITIONS FOR  
CAPTURE 54

WHERE

- CREATE MEMBER 40

WITH

- SUBSTITUTE CONTACT 51

WITH UNREGISTERED BASE TABLES

- PROMOTE REGISTRATION 28



---

## Notices

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, 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 Canada Limited  
Office of the Lab Director  
8200 Warden Avenue  
Markham, Ontario  
L6G 1C7  
CANADA

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 may contain 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 may contain sample application programs, in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (*your company name*) (*year*). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. *\_enter the year or years\_*. All rights reserved.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both, and have been used in at least one of the documents in the DB2 UDB documentation library.

ACF/VTAM	LAN Distance
AISPO	MVS
AIX	MVS/ESA
AIXwindows	MVS/XA
AnyNet	Net.Data
APPN	NetView
AS/400	OS/390
BookManager	OS/400
C Set++	PowerPC
C/370	pSeries
CICS	QBIC
Database 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/400
DB2 Extenders	SQL/DS
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	Tivoli
eServer	VisualAge
Extended Services	VM/ESA
FFST	VSE/ESA
First Failure Support Technology	VTAM
IBM	WebExplorer
IMS	WebSphere
IMS/ESA	WIN-OS/2
iSeries	z/OS
	zSeries

The following terms are trademarks or registered trademarks of other companies and have been used in at least one of the documents in the DB2 UDB documentation library:

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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.







Printed in U.S.A.

REPL-ACLP-00

