IBM

Replication Guide and Reference

Version 5

S95H-0999-00

IBM

Replication Guide and Reference

Version 5

S95H-0999-00

Before using this information and the product it supports, be sure to read the general information under Appendix C, “Notices” on page 355.

Order publications through your IBM representative or the IBM branch office serving your locality or by calling `1-800-879-2755` in U.S. or `1-800-IBM-4YOU` in Canada.

Contents

Welcome to Replication

This book describes how to plan, configure, administer, and operate the IBM Replication tools available within DB2 Universal Database Version 5 (V5) and as DataPropagator Relational Version 5.1 (V5.1) program products.

This book is for database administrators, LAN administrators, and any other knowledge workers who have responsibility for administering and operating the Capture and Apply programs, including planning, configuring, customizing control tables, defining replication sources and targets, and maintaining replication objects.

Part 1 gives an overview of IBM Replication and provides a getting started scenario.

Part 2 describes planning considerations important for designing and implementing replication.

Part 3 provides administration task information for defining and managing replication sources and targets.

Part 4 describes how to operate the Capture and Apply programs on each platform.

Part 5 provides information about mobile replication, advanced techniques, and problem determination.

Part 6 provides reference information about tables and messages.

Conventions

This book uses these highlighting conventions:

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

Terminology

This book uses industry standard terminology for database, copying, and LAN concepts, except where noted otherwise. A comprehensive glossary is located in the *DB2 Universal Database Glossary*. The following terms represent concepts specific to IBM Replication:

- **Replication**

  Refers to the propagation of data from source tables to target tables, including one-way copying, or circular copying.

- **Source server**

  The database where the source data is stored (and where the Capture program is located and executes if the database is DB2 for Windows NT, MVS, OS/400, VSE, VM, OS/2, AIX, Solaris, and HP-UX). *Source server* is synonymous with *source*.

- **Target server**

  The database where the target copy of the data is stored and where the Apply program executes. *Target server* is synonymous with *target*.

- **Control server**

  The database where the subscription definitions are located. The *control server* can be collocated at either the source or target server.

- **Full-refresh copying**

  All data is copied from the source table to the target table.

- **Update copying**

  Only changed data is copied from the source table to the target table.

- **Replication sources**

  Source tables in a replication scenario.

- **Replication subscriptions**

  Target tables and control information in a replication scenario. Also known as a *subscription set*.

- **Update-anywhere replication**

  Replication in which the replication source or the target table can be updated and the changes replicated to all related tables.

Unless otherwise noted, the following components are referred to by the following abbreviated names:

Formal name	In this book it is called ...
IBM DB2 Server for OS/390 Version 5 Release 1	DB2 for MVS V5.1
IBM DataPropagator Relational Capture for MVS, Version 5 Release 1	Capture for MVS
IBM DataPropagator Relational Apply for MVS, Version 5 Release 1	Apply for MVS
IBM DB2 Server Capture for VSE, Version 5 Release 1	Capture for VSE
IBM DB2 Server Capture for VM, Version 5 Release 1	Capture for VM
IBM Capture for the DB2 Universal Database, Version 5 Release 1	Capture for DB2 Universal Database (includes all DB2 Universal Database platforms)
IBM Apply for the DB2 Universal Database, Version 5 Release 1	Apply for DB2 Universal Database (includes all DB2 Universal Database platforms)
The replication Capture tool	Capture for *any DB2 Universal Database platform*
The replication Apply tool	Apply for *any DB2 Universal Database platform*
The Solaris Operating Environment	Solaris
DB2 for OS/2, DB2 for AIX, DB2 for HP-UX, and DB2 for Solaris	DB2 Universal Database
Capture for AIX, Capture for HP-UX, and Capture for Solaris	Capture for the UNIX** OS-based platforms
Apply for AIX, Apply for HP-UX, and Apply for Solaris	Apply for the UNIX** OS-based platforms

How to Read the Syntax Diagrams

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

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

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

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

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

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

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

- Keywords, their allowable synonyms, and reserved parameters, are either shown in uppercase or lowercase, depending on whether the platform is for MVS, OS/2, or UNIX products. These items must be entered exactly as shown. Variables appear in lowercase italics (for example, *column-name*). They represent user-defined parameters or suboptions.

When entering commands, parameters and keywords must be separated by at least one blank if there is no intervening punctuation.

- Enter punctuation marks (slashes, commas, periods, parentheses, quotation marks, equal signs) and numbers exactly as given.
- Footnotes are shown by a number in parentheses, for example, (1).
- Required items appear on the horizontal line (the main path).

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

- Optional items appear below the main path.

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

- If you can choose from two or more items, they appear vertically, in a stack.

  If you *must* choose one of the items, one item of the stack appears on the main path.

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

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

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

Road Map

If you want to ...	Refer to ...
Learn about the IBM Data Replication Solution,	Chapter 1, “The IBM Complete Data Replication Solution” on page 3.
Learn what's new in IBM Replication,	Appendix B, “What's New in IBM Replication” on page 351
Learn about replication concepts,	Chapter 2, “Introduction to IBM Replication: Tools and Concepts” on page 7.
Learn how other customers have implemented replication,	Chapter 4, “Replication Usage Scenarios” on page 51.
Plan your replication environment at the system level,	Chapter 5, “System Planning” on page 57.
Plan your replication environment at the application level,	Chapter 6, “Application Planning” on page 69.
Migrate from DPROPR V1 to IBM Replication,	Chapter 19, “Migrating from DataPropagator Relational Version 1 to IBM Replication Version 5” on page 251.

If you want to ...	**Refer to ...**
Learn about mobile replication,	Chapter 16, “Mobile Replication” on page 229.
Get started with a sample replication task,	Chapter 3, “Getting Started” on page 39.
Set replication preferences in the Tools Settings notebook,	“Setting Replication Preferences in the DB2 Tools Settings Notebook” on page 85.
Customize the replication control tables,	“Working with Customized Replication Control Tables” on page 88.
Define and manage replication sources,	Chapter 8, “Working with Replication Sources” on page 93.
Define and manage replication targets,	Chapter 9, “Working with Replication Targets” on page 103.
Configure the Capture and Apply programs,	Part 4, “IBM Replication Capture and Apply” on page 125; see the chapter for your platform.
Operate the Capture and Apply programs,	Part 4, “IBM Replication Capture and Apply” on page 125; see the chapter for your platform.
Troubleshoot errors for the Capture and Apply programs,	Part 4, “IBM Replication Capture and Apply” on page 125; see the chapter for your platform.
Learn about the target table structures,	Chapter 20, “Table Structures” on page 265.
Debug error messages,	Chapter 21, “IBM Replication Messages” on page 303 and Chapter 18, “Problem Determination Facilities” on page 241.

To locate information on other topics, see Appendix A, “How the DB2 Library Is Structured” on page 343 for a complete description of the DB2 library.

Part 1. Introduction to IBM Replication

This part introduces IBM Replication: the tools, the concepts, and how to get started using replication.

Chapter 1. The IBM Complete Data Replication Solution

IBM's complete data replication solution comprehensively delivers data replication tools that function together in a single solution that includes:

- IBM Replication: The DataPropagator Relational Version 1 (DPROPR V1) products have been updated for Version 5 (V5) of the DB2 database.
  - DB2 Universal Database V5 replication tools: the Control Center replication administration features and the Capture and Apply programs
  - IBM Capture and Apply for MVS V5.1
  - Capture for VSE and VM V5.1: integrated with IBM DB2 Server for VSE and VM V5.1
- DPROPR V1 for support of DataJoiner
- DataPropagator Relational Capture and Apply for OS/400 V3.1
- IBM DataPropagator NonRelational
- IBM DataRefresher
- IBM DataJoiner
- Lotus NotesPump

This chapter provides overview information about how these products support the complete data replication solution. This book describes IBM Replication features only. DPROPR V1, DPROPR Capture and Apply for OS/400, DataRefresher, DataJoiner, and Lotus NotesPump are discussed in other IBM publications.

Support of Replication Environments

By supporting sources and targets that include the DB2 family, IMS, VSAM, Oracle, Sybase, Microsoft, Lotus Notes, and others, IBM's solution ensures that you have timely, reliable, and consistent data across your enterprise. IBM supports heterogeneity in replication environments as follows:

- IBM's architecture is built on standard SQL to leverage the database engine capabilities for data enhancement, network connectivity, and data security. IBM database support for DRDA enables replication to other DRDA application servers.
- An architected data staging area supports interoperability among multivendor databases, between heterogeneous data models, and among products from independent software vendors. IBM's DataPropagator NonRelational and DataRefresher products bring IMS, VSAM, and flat file data into the staging area, making it available for replication to client/server targets. Additionally, independent vendors offer products that read the data staging tables relying on the IBM Capture program as the source for DB2 data.
- IBM replication products directly support multivendor sources and targets through IBM's multidatabase server product, DataJoiner. DataJoiner provides transparent

access to Oracle**, Sybase**, Microsoft SQLServer**, Informix, and other databases enabling replication to run across multivendor environments.

- Lotus NotesPump** extends replication to Lotus Notes and Open Database Connectivity (ODBC) accessible data, for example, data that can be accessed using Lotus Approach or Microsoft Access.

Replication Solution and Product Environment

Figure 1 illustrates IBM's replication solution.

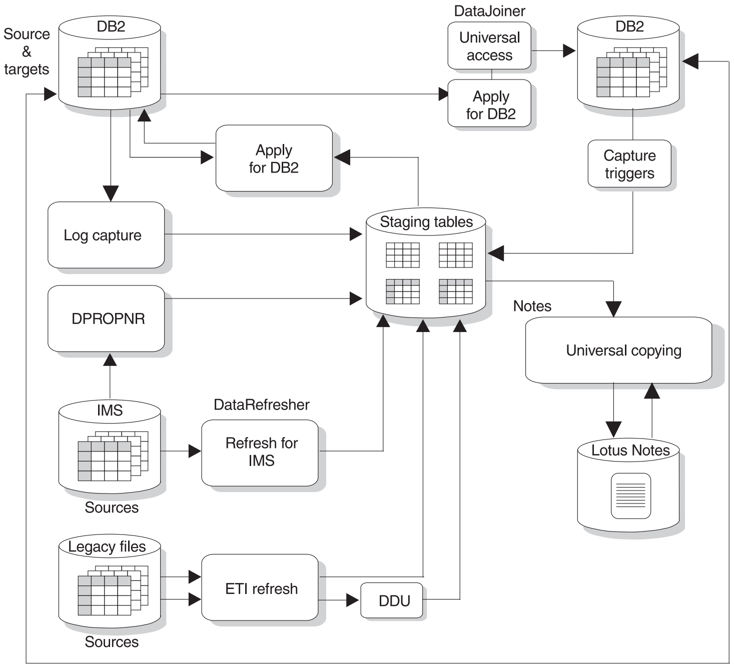


Figure 1. IBM's Data Replication Solution and Product Environment

The IBM Replication tools capture source data from DB2 and IMS. Captured data for copying is placed into staging tables before the Apply program reads the tables. Updates need to be captured only once. The updates can be subsequently staged at any number of sites, allowing for flexible data distribution.

For full refresh from relational sources, the Apply program reads directly from the relational source. Nonrelational refresh copies are copied into staging tables before being read by the Apply program.

The Apply program filters, enhances, distributes, and replicates data to the target databases. DataJoiner enables the Apply program to read from or write to multivendor sources and targets. NotesPump acts as an alternative Apply program for multivendor, ODBC, and Notes targets.

Legacy Data Sources

DataPropagator NonRelational and DataRefresher provide a coordinated replication solution for IMS data sources. DataRefresher provides full refresh processing from IMS to DB2. DataPropagator NonRelational captures changes from the IMS log and replicates them to a DB2 staging table. Replicating from IMS to DB2 makes it possible to implement decision-support applications that use relational technology. You can replicate to a staging table on DB2 for MVS, which allows IBM Replication to further filter, enhance, distribute, and apply the data to relational databases across your enterprise.

In addition to IMS sources, DataRefresher supports full refresh for VSAM and flat file sources. DataRefresher produces data extracts in load-ready format for any DB2 database and for both user tables or data staging area targets.

You can use DataRefresher's GUI to easily specify data definitions and extract requests. You can then share them with DataPropagator NonRelational to replicate changes from source to target databases.

Multivendor Data Sources and Targets

IBM's DataJoiner extends replication to multivendor sources and targets including Oracle, Sybase, Microsoft SQLServer, and others. DataJoiner is a multidatabase server that provides a new level of data access. It can help businesses meet the challenge of efficiently accessing distributed data by enabling users to develop a virtual, enterprise-wide relational database.

DataJoiner provides a single-site image of all of your data, relational and nonrelational, local and remote, from IBM and non-IBM platforms, as though the data were local. DataJoiner masks differences in SQL dialects, data access methods, networking protocols, operating systems, and so on. In the context of replication, DataJoiner provides the gateway to heterogeneous environments through DPROPR V1; the Apply program accesses multivendor sources or targets through DataJoiner as if they were all one local DB2 database.

Interoperability with Lotus Notes and ODBC Accessible Data Stores

Lotus NotesPump provides data replication between relational database management systems, ODBC accessible data stores, and Lotus Notes databases. Data can be copied bidirectionally between supported data sources.

When used alone, NotesPump refreshes the target data from a direct query of the source data. However, NotesPump Release 2 understands the IBM Replication staging area architecture, and can consequently act as an alternative Apply program connecting to multivendor, ODBC, and Notes data stores.

For example, to replicate DB2 data to Notes, you can use IBM Replication to capture the changes and create a staging table. Then, NotesPump can read the changes from the staging table and apply the changes to a Notes database.

Services

IBM and IBM Business Partners offer consulting and services supporting the IBM data replication solution. Customized services are available in addition to service offerings that help you:

- Plan and design your application.
- Install, configure, and integrate the products.
- Evaluate operational and tuning considerations.
- Evaluate application and data migration.
- Educate and train staff.

Contact your IBM software provider, or call in the U.S. and Canada, 1-800-IBM-3333, for additional information on IBM products and services.

IBM Replication Education

IBM Replication classes are provided by IBM Education and Training.

On the Web: IBM* Education and Training information is available on the World Wide Web. To access the entire curriculum of courses directly, the IBM Global Campus URL is **http://www.training.ibm.com/ibmedu**.

Custom Classes: Replication courses can be tailored to address your unique environment and needs. To find out more information, call 1-800-IBM-TEACH, Ext. CUSTOM (800-426-8322, Ext. CUSTOM).

IBM Employees: For complete course descriptions, see the EDUCATION application on HONE or MSE.

Chapter 2. Introduction to IBM Replication: Tools and Concepts

This chapter:

- Describes how the replication tools work.
- Describes concepts unique to replication.

The Replication Tools

IBM Replication is a set of easy-to-use, automated copy tools that replicate relational data from sources to targets. Data can be copied:

- Among DB2 for MVS, DB2 for VSE and VM, DB2 Universal Database, and DB2 for OS/400 database servers
- Across platforms for MVS, VM, VSE, OS/400, AIX, HP-UX, OS/2, Windows NT, Windows 95, and the Solaris operating environment
- On non-DB2 sources and targets through DataJoiner; such as Oracle, Sybase, Informix, Microsoft SQLServer.

You can use IBM Replication to define, synchronize, automate, and manage copy operations from a single control point for data across your enterprise, as shown in Figure 2.

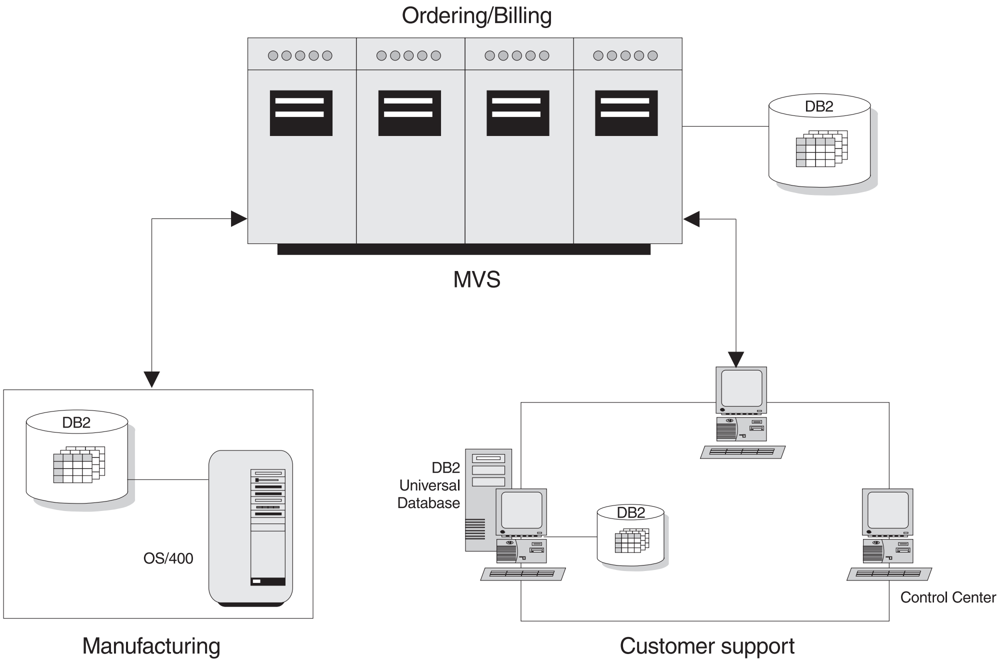


Figure 2. IBM Replication Tools. The IBM Replication tools copy data across your enterprise.

You can tailor or enhance data as it is copied and deliver detailed, subset, summarized, or derived data when and where it is needed. IBM Replication consists of the replication administration features of the Control Center and two tools, the Capture and Apply programs. The Control Center provides administration support for the replication environment with objects and actions that define and manage source and target table definitions. The Capture and Apply programs are responsible for capturing the updates to the source tables and applying the changes to the target tables.

The following sections describe how the IBM Replication tools work interact with each other. Replication concepts related to each tool are also explained.

The Control Center

The Control Center is the database administration tool for the DB2 Universal Database and includes replication administration. The Control Center automates many initialization functions, such as creating target tables and control tables when you create your replication subscriptions. A replication administrator performs the replication administration tasks.

Administering Replication: Overview

You work with three GUI objects, **Tables**, **Replication Sources**, and **Replication Subscriptions** to manage replication sources and targets.

Specific Control Center administration tasks include:

- Defining tables as sources, called *replication sources*
- Removing replication sources that are no longer sources for copies.
- Defining target table definitions, called *replication subscriptions*.
- Cloning replication subscriptions to other servers.
- Removing replication subscriptions no longer in use.

How the Control Center Manages the Replication Environment

The Control Center helps you perform two major tasks: defining replication sources and defining replication targets, or replication subscriptions. Replication sources are DB2 tables, or views, used as sources for copying data to a target table. Replication subscriptions are the specifications for one or more target tables and their location, structure, and timing schedule, as well as any necessary SQL enhancements. Other replication tasks include maintaining the sources and targets, and cloning them to other servers.

When you use the Control Center to perform replication administration tasks, it creates and maintains the replication control tables used by the Capture and Apply programs. These tables are created and stored at the source, control, and target servers. Thus, the workstation where the Control Center is located must have connectivity to all the databases where source tables exist and the Capture program runs (the source server), where the target tables will be created and the Apply program usually runs (the target server), and where the subscription control tables are stored (the control server). The exception is the "occasionally connected," or mobile environment, in which you can create and save as SQL a replication subscription definition, transfer it to another machine, and run it from there.

The Control Center is not used to operate the Capture and Apply programs.

Replication Sources and Targets: You can access your replication source and target definitions through the Control Center. There are three containers that organize the objects that you use to set up and maintain your replication environment:

Tables folder
Contains user-defined DB2 relational tables, as well as system catalog tables. You select the tables in the folder to define replication sources.

Replication Sources folder
Contains tables that have been defined as replication sources. These can be DB2 tables, system tables, views, or target tables redefined as sources for replication.

Replication Subscription folder
Contains replication subscription definitions for copying updated source data to target tables.

Figure 3 shows the Control Center window.

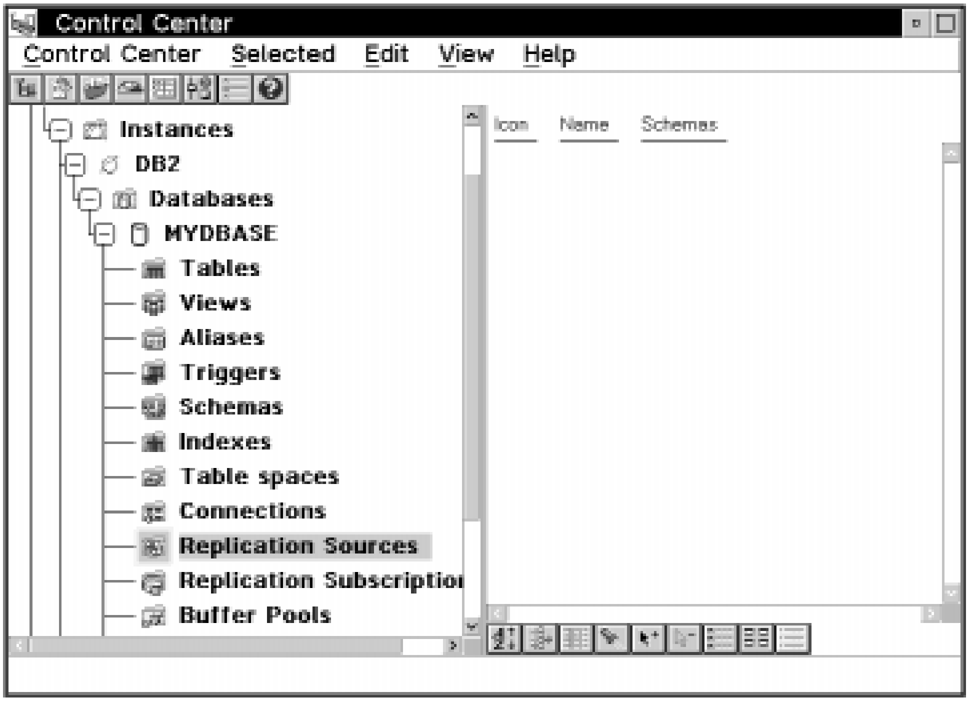


Figure 3. The Control Center and the Replication Folders. The Control Center and the replication folders provide administration tools for replication.

See “Capacity Requirements Analysis” on page 63 and “Authorization Requirements for Administration” on page 66 for information about administration planning. See Part 3, “Administering Your Replication System” on page 81 for information about administration tasks. See *DB2 Getting Started* for your platform for more information about the Control Center.

The Replication Control Tables: IBM Replication tools use control tables to communicate with each other and to manage replication requests such as defining and managing sources and targets, capturing changes, and replicating changes. These control tables are located at the source, control, and target servers, which are defined in “Terminology” on page xv.

The replication control tables are created in one of the following ways:

- By customizing and running the DPCNTL.* files.
- The first time you define a replication object at a database server.

Most customers need to customize the tables for their site requirements or DB2 platform. To customize the control tables, you *must* edit the DPCNTL.* files and run then *before* performing any replication requests, such as defining replication sources, at the source or control server. See “Working with Customized Replication Control Tables” on page 88 to learn how to customize and run the DPCNTL.* files.

SQL Generated for Replication Requests: Whenever you submit a replication request from the Control Center, such as defining a replication source, the processing information is generated as SQL statements. You can choose to run the statements immediately, or you can save the SQL statements to a plain ASCII file, which you can edit and run at a later time from a replication folder object in the Control Center.

Deferred SQL files allow you to customize the replication tasks for your shop or application and give you flexibility as to when and how you run the SQL files. You can batch the definitions together, defer the processing of the replication action until a specified time, and create libraries of SQL files for backup, site-specific customizing, or to run stand-alone at distributed sites, as in a mobile environment. You can rerun the definitions as necessary. See “Customizing and Running Replication SQL Files” on page 89 to learn how to defer and customize SQL files.

The Capture Program

The Capture program is the replication tool that captures the changed data and makes the changed data available for replication. It runs at the source server database.

Capturing Changed Data: Overview

The Capture program captures changes made to data in replication source tables by reading the database log or journal. It places the captured changes into change data tables. There is only one Capture program at each domain of the database directory, that is, one Capture program in each of the following:

- DB2 Universal Database database
- DB2 for MVS subsystem in a non-data-sharing environment

- DB2 for MVS data-sharing group
- DB2 for VM or DB2 for VSE database

The Capture program usually runs continuously, but you can bring it down while running utilities or modifying replication sources. The Capture program runs independently of the Control Center, but uses control information created by the Control Center.

Tasks for the Capture program include:

- Starting
- Stopping
- Suspending temporarily
- Reinitializing
- Specifying tuning parameters
- Pruning

See the Capture and Apply chapter for your Capture program platform for instructions about performing these tasks.

How the Capture Program Captures Changes

DB2 records every transaction in a log file for diagnostic and recovery purposes. The Capture program monitors the DB2 log to detect change records from source tables that both have the DATA CAPTURE CHANGES attribute and are defined as replication sources. The Capture program retrieves change and commit information from the active and archive logs on DB2 for MVS 4.1 or higher and DB2 Universal Database. [1] These records contain a before-image and after-image of the table row. The Capture program captures these changes in the CD tables. The Capture program also maintains information about committed units of work in the unit-of-work (UOW) table. This table is joined with the CD table to identify and replicate committed updates. The Apply program can then read the CD table and copy the changes to the target site and apply them to copies of the source table. Figure 4 on page 12 shows the relationship between the Capture program, the DB2 log, the source table, and the control tables.

[1] Capture for MVS can only retrieve the active log on DB2 for MVS 3.1. Capture for VSE and VM 5.1 can read only the active log on DB2 for VSE and VM.

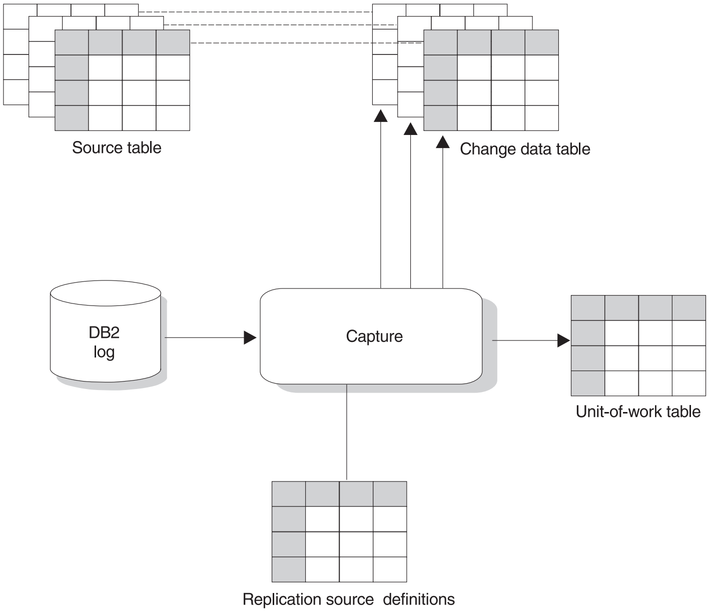


Figure 4. The Capture Program at the Source Server. The Capture program at the source server reads the DB2 log and captures changed data to the CD table.

Updates made to source tables are written to the log when the DB2 page is full. (A DB2 log page is usually 4 KB long.) The Capture program captures changes from this log.

Control Tables Used by the Capture program: The Capture program uses control tables to manage capturing changes from multiple source tables. These tables are created while setting up administration for the Control Center. The control tables used by the Capture program are located at the source server. The following list describes the source server control tables:

Register
Contains the name of the source table and the names of the change data table associated with the source table.

Critical section
Contains the Apply qualifier and is used for concurrency control between the Capture and Apply programs.

Tuning parameters
Contains performance tuning information specific to the Capture program.

Trace
Contains trace information for the Capture program.

Unit-of-work (UOW)
Contains committed unit-of-work information used by the Apply program.

Pruning control

Contains information about how far through the CD table the Apply program has progressed in replicating changes to the target tables. The Capture program uses this information to prune the CD table.

Warm start

Contains information about the units-of-work in progress and starting sequence information necessary for restarting the Capture program without performing a full refresh.

Change data (CD)

Contains the changed data read by the Capture program from the database log or journal, and is a source for update copies. There is only one CD table for each replication source table.

These tables are described in more detail in Chapter 20, “Table Structures” on page 265.

How the Change Data and UOW Tables are Pruned: The Capture program begins capturing changes from the DB2 log when requested by the Apply program. The Capture program puts the changes into the CD table; there is one CD table for each defined replication source that has DATA CAPTURE CHANGES enabled. Additionally, the Capture program records the unit of work information in the UOW table.

These two tables have the potential for unlimited growth; therefore, it is important to manage the number of rows to be kept by discarding rows that are no longer needed. This process is called *pruning*.

Replication Logical Partitioning Key Support: By default, the Capture program captures an update to the source table as an UPDATE statement. In cases a source column change updates a target primary key column, a column in the predicate for a target table, or to a table space partitioning key column, you can have the Capture program capture the updates as DELETE and INSERT statements. This is known as replication logical partitioning key support. You must explicitly set this option while defining replication sources.

Without replication logical partitioning key support, when the primary keys of either the source or target tables are being updated, the Capture program captures the changed row for the update. The Apply program then attempts to make an update to a row on the target table with the new key value. This new key value is not found in the target table, so the Apply program converts this update to an insert. The new row is fine, but the old row with the old key value remains in the table and is unnecessary. When you enable replication logical partitioning key support, the Capture program captures the change as separate DELETE and INSERT statements in the CD table.

For example, the primary key value for a table is the employee last name "Wood" and the employee gets married and changes her last name to "Smith". Without replication logical partitioning key support, when the Apply program updates the employee's last name with a new primary key value, it does not find the value "Smith" and inserts a new row. The row with the primary key value "Wood" is not deleted.

With replication logical partitioning key support, the row with the primary key value "Wood" is first deleted, and a new row with the primary key value "Smith" is inserted.

See “Replication Logical Partitioning Key Considerations” on page 72 to learn when to use replication logical partitioning key support. See “Defining a Custom Replication Source” on page 95 to learn how to specify that updates are captured as DELETE and INSERT statements.

Capture for MVS in a Data-Sharing Environment: The Capture program requires exclusive use of the IBM Replication control tables in a subsystem. In a data-sharing environment, tables are shared among multiple subsystems. Consequently, if you are using DB2 for MVS/ESA Version 4.1 or higher, only one instance of the Capture program can run for all subsystems that are members of a data-sharing group.

The Apply Program

The Apply program is the replication tool that replicates copies of the source table data to the target table. You can run it at any server but it is generally run at the target server database.

Applying Changed Data to Target Tables: Overview

The Apply program reads the changed data previously captured and stored in a CD table and applies the changes to target tables. The Apply program also reads data directly from source tables when copying the entire source table data for a full refresh to the target table.

The Apply program generally runs at the target server, but it can run at any server in your network as long as it can connect the source, control, and target servers. Several Apply program instances can run on the same or different servers.

The Apply program runs independently of the Control Center, but uses control information that the Control Center creates. The control information that the Apply program uses is stored in tables at the control server.

See “Types of Copies” on page 17, and Chapter 20, “Table Structures” on page 265, for more information about target table types.

Tasks for the Apply program include:

- Starting
- Scheduling
- Stopping

See the Capture and Apply chapter in this book for your Apply program platform for instructions about performing these tasks.

How the Apply Program Replicates Data

When the Apply program reads the changed data stored in CD tables, it applies it to target tables at either local or remote servers. It can also apply column functions, such as SUM and AVG, to the source table or CD table and append the result to the target tables. The Apply program can run at any server that can connect through the SQL CONNECT statement to each database server where source and target tables reside. Figure 5 on page 15 shows the Apply program's relationship with the source server control tables, the subscription definition control tables, and the target table. The control tables used by the Apply program are described below.

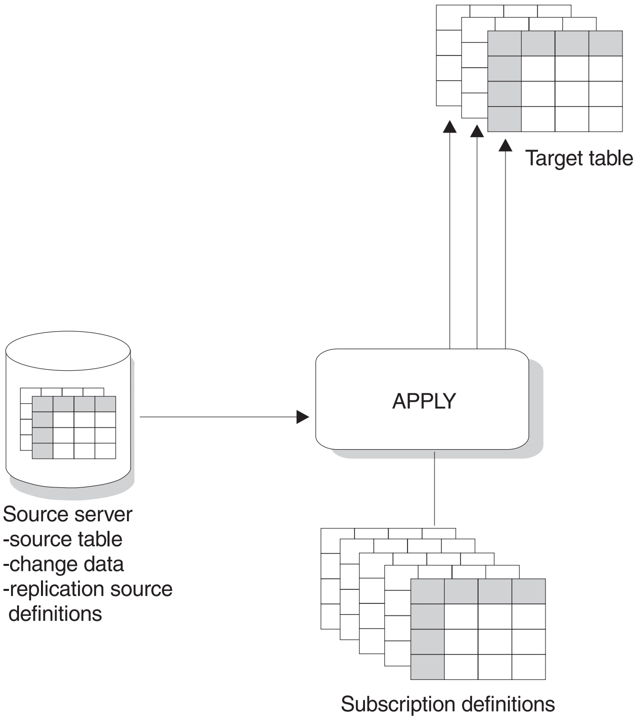


Figure 5. The Apply Program. The Apply program reads the source server tables, manages the subscription control tables, and updates the target tables.

The Apply Qualifier: When the Apply program is started, it is supplied with an arbitrary job qualifier independent from a logon user ID. Known as the Apply qualifier, it is associated with individual replication subscriptions and is responsible for the replication of only those replication subscriptions.

The Apply qualifier allows a user ID to run more than one instance of the Apply program, using different Apply qualifiers.

Control Tables Used by the Apply Program: The Apply program relies on the following control tables to control the replication of source table changes to the target tables:

Source server control tables
The control tables used by the Capture program.

Control server control tables

Apply trail
Records a history of refreshes and updates performed against target tables.

Subscription set
Defines the characteristics of the replication subscription, such as the replication subscription name, the owner of the replication subscription, the Apply qualifier that is associated with the replication subscription, and the timing of the replication subscription.

Subscription statements
Contains the SQL statements or stored procedures to be run before or after the replication subscription is processed.

Subscription events
Contains the timing information for copying the replication subscription based on event triggering. This table is maintained by user application.

Subscription targets members
Identifies each source-to-target pair in the replication subscription.

Subscription columns
Contains supplemental information for the replication subscription control table, such as renamed columns or computed columns.

These tables are described in more detail in Chapter 20, “Table Structures” on page 265.

The control tables used by the Apply program reside in the source server and the control server.

The Control Server: The control server is the logical server that contains the subscription control tables. Each Apply program is associated with a control server, which is specified during invocation of the Apply program. Multiple Apply program can share a control server.

The control server can be located at the source server, the target server, or any database server that the Apply program can connect to. For better performance, the control server should be located at the server where the Apply program runs because the Apply program frequently reads the tables in the control server. However, locating the control server at the source server, if it is a mainframe database, can provide improved security.

Each control server is associated with one or more Apply qualifiers. The qualifier associates a control server with an Apply program and a replication subscription. This association allows a replication subscription to be serviced by one Apply program and identifies where the control tables for that subscription are located. You cannot easily change the Apply qualifier, so it is important to plan carefully.

Types of Copies: You can define a replication subscription with the Control Center to create the following types of copy tables, known as target table types:

User copy
A complete, condensed copy of the replication source table that must have a primary key.

Point-in-time
A complete, condensed copy of the replication source table at a certain point in time that must have a primary key. This table contains timestamp columns to indicate when a transaction occurred.

Base aggregate
A history table in which new rows are appended for each subscription cycle using the result of an SQL column function calculation against the replication source table data.

Change aggregate
A history table in which a new row is appended for each changed row in the replication source table using the result of an SQL column function calculation against only recently changed data.

Condensed, noncomplete consistent change data (CCD)
A small staging table that, when defined locally to the source, is useful for netting out “hot spot” updates before replication to other sites.

Condensed, complete CCD
A full-sized staging table useful for efficient remote staging, which allows for remote copies to be both initialized and maintained without needing to access the original source each time it is updated.

Noncondensed, noncomplete CCD
A table that is initially empty and is appended by each insert, update, and delete action; useful for auditing purposes.

Noncondensed, complete CCD
A table that starts out as a copy of the full-sized source table and is appended by each insert, update, and delete action. It retains all information about the source table and supports “as of” historical queries. For example, it can return an answer to a query as if you had run the query against the replication source table last Tuesday, or a month ago, or yesterday.

Replica
A target table that can be updated. Changes to this table are replicated back to the replication source table. This table is used in update-anywhere scenarios.

The Apply Processing Cycle: In the more common "pull" mode, the Apply program runs at the target server and connects to the source and control servers to pick up changed data and to read the control tables. Table 1 on page 18 describes how the Apply program completes a replication cycle from the target server.

Table 1. The Apply Program Cycle: A high-level overview of how a replication subscription is processed.

Step	Server
1. Look for work; check the subscription control tables.	Control server
2. Pick up recent change data to be applied to the target table.	Source server
3. Write the answer set into a local "spill" file (possibly an in-memory file).	Target server
4. Apply the change data in the spill file to the target table.	Target server
5. Update subscription status.	Control server
6. Report subscription progress in the pruning control table.	Source server

Full-Refresh and Update Replication: The Apply program copies data from the source to the target either by refresh or update copying. In refresh copying (also known as full refresh), the Apply program copies the entire source table and copies it to the target table. The Capture program does not capture changes, and there are no CD or CCD tables involved. You can specify refresh copying while defining the replication source table. If you have large tables, you might want to use a fastload program to simulate an initial full refresh copy. See the Capture and Apply chapter for your platform in this book for more information about fastload programs such as ASNLOAD.

In update copying, the Apply program copies only the changed data from the CD table to the target table. The first time the Apply program copies data to the target table or after a cold start of the Capture program, the Apply program uses refresh copying to populate the target table (See the Capture and Apply chapter for your platform in this book for more information about cold start). After the target table is populated, update copying is used. This type of copying is the default and is used unless you specify otherwise while defining the replication source. In order to capture changes for a replication source, the DATA CAPTURE CHANGES clause of the CREATE or ALTER TABLE SQL statement must be enabled. The Control Center automatically alters a replication source table for this option, unless you specify refresh copying or the table is ineligible for update copying.

Subsetted Target Tables: IBM Replication supports both vertical and horizontal subsetting of the source table. This means that you can specify that only a subset of the replication source table columns and rows are replicated to the target table, rather than all of the columns and rows.

Vertical Subsetting: In some replication scenarios, you might not want to replicate all columns into the CD table. With vertical subsetting, you can define the CD table as having fewer columns than your base table. This is appropriate if the column data types are not supported.

Attention: Vertical subsetting is not available for replica tables.

You can define vertical subsetting in two ways:

- While defining a replication source table, using update copying: Deselect the columns that you do not want *available* for replication to a target table. The CD table does not contain the columns that you deselected.

Note: CD tables must contain sufficient key data for point-in-time copies, which are maintained with key-qualified predicates.

- While defining a replication subscription: Use the advanced subscription options to deselect source table columns that should not be subscribed to for the target table.

Horizontal Subsetting: You can also subset the target table horizontally by specifying a row predicate while defining the replication subscription for the target table. Use the advanced subscription options to define a WHERE clause.

Target table primary keys are assumed to be invariable. If the primary key of the source table is updated, the Capture program places an UPDATE row in the change data table for the source table. When Apply reads this UPDATE row, it first attempts to perform a searched UPDATE, which will likely fail with a "row not found" condition. Apply then automatically reworks the failed update into an INSERT. The target table then can have both the original row with the original key *and* the updated row with the new key. The original key no longer exists in the source table.

Attention: If one or more of the columns defined in the predicate are updated, you must use the Capture program function that defines updates as DELETE and INSERT statements. See “Defining a Custom Replication Source” on page 95 to learn how to use this option.

Defining New Columns in the Target Table: One of the advantages of IBM Replication is that it allows you to create new columns in the target tables with the exact information from existing columns in the source table. You can also create a new column in the target table based on:

- The computation of one or more existing columns in the source table
- Other SQL functions that summarize or average all the rows from a source table (or CD table in the case of a change aggregate table) into a single summary row

A target table containing any aggregate columns with other SQL functions such as AVG and SUM is an *aggregate table*.

Advanced Replication Concepts

The following section describes concepts unique to IBM Replication.

Replication Subscriptions: Ensuring Data Consistency

A replication subscription defines the source-to-target relationship between the replication source and a target table. It also defines the structure of a target table and how the Apply program replicates data to the target table. A replication subscription can contain one or more of these definitions, called members. A replication subscription can be used to group one or more members for consistency. Changes are replicated from all replication source tables to the target tables by replicating the changed data for all the members in one subscription cycle; one unit of work at the target server.

Because the replication subscriptions have something in common, the changed data needs to be copied in the same unit of work. Each replication subscription has a name and is identified by a row in the subscription set control table. A replication subscription with replica tables has two rows, one for each direction of copying (from source to replica and replica to source).

Figure 6 shows the relationship of the replication subscription with the Apply program.

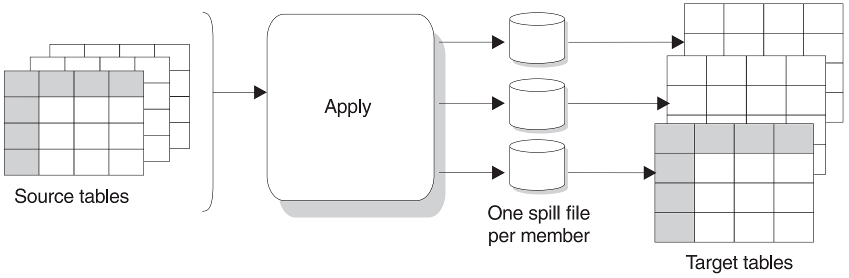


Figure 6. Replication Subscriptions and the Apply Program

The Apply program reads changes from the replication source or CD table and creates one spill file created for each table. The Apply program replicates the changes to the target tables.

See “Data Consistency Requirements” on page 76 for more information about replication subscription rules and recommended uses.

Before-Image and After-Image Columns

A before-image column is a copy of a column before it is updated. DB2 logs both the before-image and after-image columns of the table for each change to the table. A before-image copy is useful in some industries that require auditing or application rollback capability.

Before-Image Columns Values in the Target Table

When you specify to capture before-image columns, the before- and after-image columns in the target table have the following values when the source column changes:

Type of Change	Column Value
Insert	The before-image column has a NULL value.
Delete	Both the before- and after-image columns contain the before-image value.
Update	Column values before the change are captured in the before-image columns; values after the change are in the after-image columns.

NULL Values in Before-Image Columns

When the target table is initialized (or full refreshed due to a Capture program cold start), before-image columns have NULL values. Full refresh copies are from the source table, so no before-image values are available. For example, the before-image columns of a point-in-time target table have NULL values if the replication source table has no updates because no before-image columns were copied.

More about the Target Table Types

IBM Replication offers you many choices for target table structures. Depending on the kind of data you want in your application system, you can choose how changes are copied to the target table. Target tables function as historical or trend information, sources for update-anywhere replication, or simply an identical copy of the source table.

User copies

A user copy table reflects a valid state of the source table, except for subsetting and data enhancement, but not necessarily the most current state. Because the tables are physical objects, references to user copy target tables (or any other target table type) reduce the contention that results from too much direct access to the source tables. Accessing local user copy tables is much faster than using the network to access remote source tables for each query.

Point-in-Time Tables

A point-in-time table contains an added system column, the approximate timestamp of when the particular row was inserted or updated at the source system. Otherwise, a point-in-time table is much like a view of the source table, but at some time in the past. Point-in-time copies reflect a valid state of the source table, but not necessarily the most current state.

Aggregate Tables

Queries with column functions (AVG, MAX, MIN, SUM, COUNT) are very useful. They can range over many thousands of qualifying rows, yet they return very compact, easy-to-understand results.

By adding a timestamp and aggregating the results of these calculations, you can track broad trends, while still retaining the benefits of data reduction and compact storage.

Aggregate tables are built when rows are added or appended to the table over time.

Base Aggregate Tables In this table, calculations are made against the source table. This type of history is useful for tracking broad indicators that have relatively low volatility. Base aggregate tables can include calculations against stable base data; calculations are made according to the subscription frequency criteria.

Change Aggregate Tables In this table, calculations are made against change data, not base data. Each calculation ranges over the recent changes since the time of the last calculation. You can track insert, update, and delete operations collectively or individually, determined by the filtering predicates that you specify. If there are no recent updates, calculation is deferred, creating very compact storage.

Replicas

Replica tables are very similar to user copy tables, except that replica tables are the only target type that an application can update and have its changes replicated back to the replication source table. A replica is a source of updates to the replication source and other replicas.

Auto-Registration

Some target table types are designed to become sources for additional replication. The CCD table and the replica table are almost always used in multiple tier or circular replication scenarios and are automatically defined as replication sources after they have been defined as part of a replication subscription.

After you define these tables as replication sources, they are immediately available as sources for additional replication subscriptions.

SQL Statements and Stored Procedures

You can define run-time processing statements using SQL statements and stored procedures before and after the Apply program processes the replication subscription. This feature is useful for pruning CCD tables and controlling the sequence in which replication subscriptions are processed. The run-time processing statements can be run at the source server before the replication subscription is processed, and at both the source and target servers after the replication subscription is processed. The stored procedures use the SQL CALL statement, newly supported by IBM Replication, without parameters. The run-time procedures are executed together in a single unit-of-work. Acceptable SQLSTATEs can be defined for each processing statement as well.

Data Blocking: Using MAX_SYNCH_MINUTES for Large Jobs

Large amounts of data to be replicated often need to be broken down into more manageable sizes to prevent spill file or log overflows. For example, following an extensive Apply program outage, if a large block of change data has accumulated in the CD tables, breaking down the block of data prevents spill file overflows if you have limited disk space for the spill file.

You can use the blocking factor to specify how many minutes worth of change data can move in a subscription cycle. If the accumulation of change data is greater than the number of minutes specified, the Apply program converts a single subscription cycle into many mini-cycles, reducing the backlog to manageable pieces. This reduces the stress on the network and DBMS resources and reduces the risk of failure. If a failure occurs during one of the mini-cycles, then only the failed mini-cycle needs to be redone. When the Apply program later retries the replication subscription, it will resume from the point following the last successful mini cycle. Figure 7 on page 23 shows how the changed data is broken down into subsets of changes.

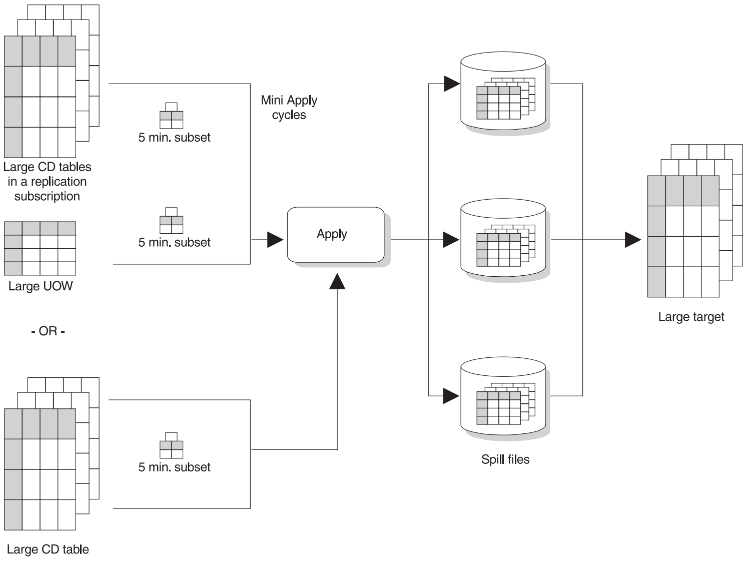


Figure 7. Data Blocking Value. You can reduce the amount of data being replicated at a time by specifying a blocking value.

The column MAX_SYNC_MINUTES in the ASN.IBMSNAP_SUBS_SET control table contains the value you specify and uses it as a time-threshold limit that regulates the amount of changed data that the Apply program fetches and applies during a given subscription cycle. The Apply program uses this value to determine the interval upper bound on the amount of data to copy. When it reaches the upper bound, the Apply program ends the cycle.

Replication Subscription Timing

The Apply program determines when to start a subscription cycle based on the information you provide while defining the replication subscription. You can set an interval or relative timing schedule, or trigger the Apply program with events to start processing a replication subscription.

Relative timing

Relative timing is copying on a specified interval. You determine a specific start time, date, and interval, which can be a set number of minutes or continuous copying. Relative time intervals are approximate. The Apply program will begin processing the replication subscription as soon as it can, based on its work load and the availability of resources.

Event timing

Event timing is copying that is triggered by an event. You specify an event name that will trigger the Apply program and have an application populate the event table with a timestamp. When the Apply program reads the event control table and the time has lapsed, it begins processing the replication subscription.

The event control table, ASN.IBMSNAP_EVENT has three columns as shown in Table 2 on page 24.

Table 2. The Event Control Table

EVENT_NAME	EVENT_TIME	END_OF_PERIOD
END_OF_DAY	1997-12-01-17:00:00.000000	1997-12-01-15:00:00.000000

EVENT_NAME is the event that you specify while defining the replication subscription. The Control Center populates this column after the replication subscription has been defined. EVENT_TIME is the timestamp for the time when the Apply program begins processing the replication subscription. END_OF_PERIOD is an optional value that indicates that transactions after this time should be deferred until a future date.

In Table 2, END_OF_DAY is the event name that the administrator specified while defining the replication subscription. The timestamp value, 1997-12-01-17:00:00.000000, is the time when the Apply program is to begin processing the replication subscription. The timestamp value, 1997-12-01-15:00:00.000000, is the transaction time when changes are no longer replicated; these transactions will be replicated on the next day's business cycle.

Your applications post events, not the Control Center. You tie your applications to subscription activity through named events. If you post an entry using CURRENT TIMESTAMP for EVENT_TIME, then you trigger the event named by EVENT_NAME. Any replication subscription tied to this event is now eligible to run. If you like, you can post events in advance, such as next week, next year, or every Saturday. As long as the Apply program is running, the Apply program will start at approximately the time that you specify. If the Apply program is stopped at the time you specify, and then restarted later, it checks the event table, notices the posted event, and begins processing the replication subscription.

Staging Changed Data

One of the advantages of IBM Replication is that it allows you to *stage* changed data; that is, the Capture program captures changes to a source table only once and inserts change data rows into a CD table. The Apply program then pulls the changes from the CD tables. The Capture program also automatically prunes change data rows from change data (CD) tables when they are no longer needed; it does not, however, prune change data from consistent change data (CCD) tables.

CD and CCD Tables

A CD table receives an arbitrary number of change data rows from the Capture program that are not condensed. The CD table has no knowledge of transaction boundaries, or whether the transactions issuing the updates are committed, are incomplete, or are in flight. The Apply program joins the CD tables with the unit-of work (UOW) table to determine the committed changes to apply to copies. Uncommitted changes are eventually pruned, depending on the retention limit that you define in the Capture program tuning parameters control table.

Rows in a CD table reflect changes that are equivalent, if not identical, to the original operational updates. Uncommitted and incomplete changes can appear in rows in a CD table.

CCD staging tables are copies, defined in much the same way as point-in-time copies. They are the join of a the CD and UOW tables and contain only committed change data. Figure 8 shows which columns of the UOW table are included in the CCD table.

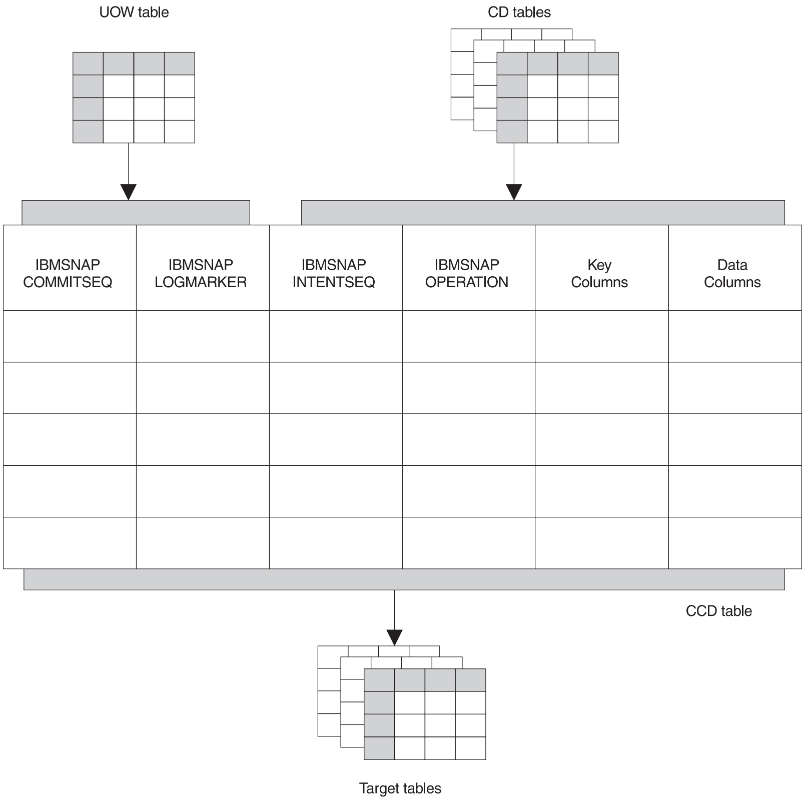


Figure 8. The CCD Table. This table is the join of the UOW table at the source server and the CD table for the replication source table.

Table 3 on page 26 shows the options that you have when you define these staging tables, and which options are the default selected by the Control Center.

Table 3. CCD Staging Table Attributes

CCD Is Local to:	Is CCD Complete?	Is CCD Condensed?	How Can CCD be Used in This Configuration?
Source table	Y[1]	Y	The CCD table is redundant with the local user table, and is therefore not usable as a history.
Remote copies	Y[2]	Y	As a remote staging table, where history retention is not required.
Source table	N[2]	Y	As a local staging table, which provides a stable source for synchronizing “fan out” copies.
Remote copies	N[1]	Y	By advanced users, who need to write their own apply programs to maintain indexed files or foreign DBMS tables. This configuration cannot support initialization of new point-in-time copies.
Source table	Y[1]	N	As local history table.
Remote copies	Y[2]	N	As remote history table.
Source table	N[2]	N	As a general-purpose, local staging table. When used with the original user table, this configuration can support copies that are complete histories. This table contains all change data for a given interval, but is not a complete history without the user table.
Remote copies	N[1]	N	By advanced users, whose application requirements are for change data only. This configuration cannot support initialization of point-in-time copies.

[1] Table and definitions must be set up outside of the Control Center.

[2] The Control Center creates the table and sets up definitions.

Staging Tables

CCD staging tables hold captured changes from insert, update, or delete operations against a base table. CCD staging tables can be local to, or remote from, the original source.

A *complete* staging table contains every row of interest from the original source. A *condensed* staging table contains only the most current value for the row. Because condensed CCD tables do not have the same potential for unlimited growth as CD tables, they do not usually require pruning. Noncondensed CCD tables contain complete histories.

Apply inserts rows into noncondensed CCD tables; for condensed CCD tables, Apply updates rows already in the table. Apply uses CD and CCD tables as sources when maintaining replication subscriptions for point-in-time, change aggregate, and CCD target tables.

Benefits of Staging Data

You can use staging tables to:

- Maintain complete histories of data changes.
- Design generalized information and data warehouses.
- Support a variety of data delivery configurations.

- Minimize change data and unit-of-work join processing.
- Condense “hot spot” updates before transmitting data.

For example, you can use staging tables as part of a replication scenario that includes data from IMS and other sources. IBM's DataPropagator NonRelational can deliver IMS change data into a CCD staging table. You can then define the CCD table as a replication source.

With staging tables, you can set up sophisticated distribution networks and balance your work load across multiple DBMSs. You can copy DB2 for MVS changes to a DB2 Universal Database database, and then have, for example, 50 other replication subscriptions referring only to the staging table. In this way, the DB2 for MVS database is required to maintain only one set of copies directly, although dozens of copies are maintained indirectly.

Using Internal, Local, and Remote CCD Tables: When you create the first CCD table in a replication subscription of a source table and store the target table at the same source server, the target table is called an *internal CCD table.* Internal CCD tables are created by joining the unit-of-work and CD tables locally. They serve as a local cache for committed changed data.

If you create a second CCD table replication subscription against that source table, and store the target table at the same source server, the target table is called a *local* CCD table; only the first CCD at the source server is the internal CCD. Figure 9 on page 28 shows an example of an internal or local CCD table that replicates changes to two target tables.

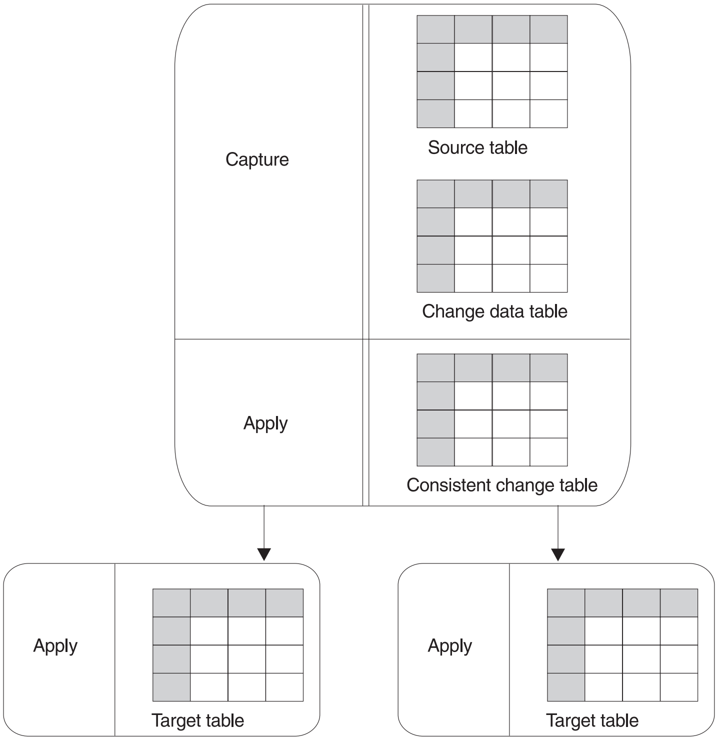


Figure 9. An Internal or Local CCD Table. The internal or local CCD table is located at the source server.

A CCD table at a server which is not the source server is called a *remote* CCD table. The remote CCD table can be located at the target server or an intermediate staging server as shown in Figure 10 on page 29.

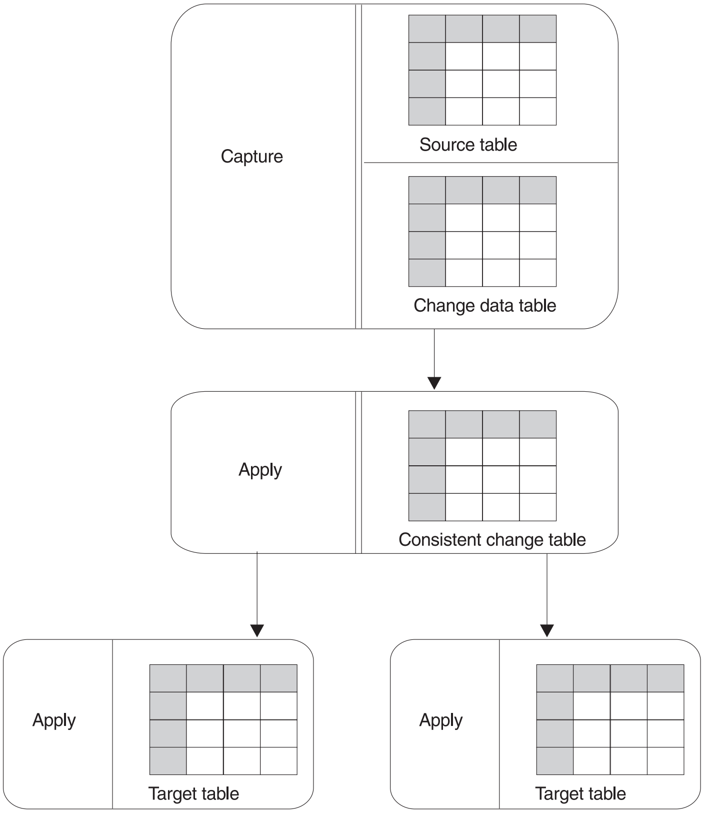


Figure 10. A Remote CCD Table. The remote CCD table is located at an intermediate staging server.

When you subscribe to a source table, the Control Center automatically defines certain target tables as a replication source for further copying, known as auto-registration. However, internal CCD tables are auto-registered differently than CCD tables. When internal CCD tables are auto-registered, replication source information is stored in the CCD_OWNER and CCD_TABLE columns of the source table row in the register control table. Staging tables created in subsequent replication subscriptions are defined as replication source tables.

This difference in auto-registration means that you cannot create replication subscriptions against internal CCD tables. Instead, you create replication subscriptions against the replication source table. The Apply program follows the hierarchy described in "How the Apply Program Selects a Source Table" on page 31 to select a source table from which to copy. In most cases, the Apply program chooses the CCD table associated with the source table.

External Data Sources: Changes captured within applications or other system tools, such as DataPropagator NonRelational, can also be defined as sources for replication subscription. The external data source must provide a complete CCD table and the

CCD table must be updated by the application. For example, if an IMS segment is the source, DataPropagator NonRelational updates the DB2 CCD table. If the source table is not from IMS, you will need another program to update the CCD table. You can then define the CCD table as a surrogate replication source table with the Control Center. The CCD table can be stored and defined as a replication source in any supported database. You can then define replication subscriptions, regardless of whether the original transaction updates occurred in an IMS or DB2 database.

Transaction-Based Versus Transaction-Consistent Replication: Using Internal CCD Tables to Reduce Network Load

IBM Replication supports both transaction-based replication (replication of every update used by every transaction) and nontransaction-based replication (replication of just the net results of the recent activity).

The following example illustrates the difference between the two types:

```
Transaction 1: Update table1 set col1 = 'X' where key1 = 425
               Update table2 set col2 = 'B' where key2 = 425
Transaction 2: Update table1 set col1 = 'Y' where key1 = 425
Transaction 3: Update table1 set col1 = 'Z' where key1 = 425
```

In transaction-based replication, all four transactions are captured and replicated. In transaction-consistent replication, only the second update in Transaction 1 and Transaction 3 are replicated.

Transaction-based replication is necessary in update-anywhere scenarios.

Transaction-consistent replication is superior to transaction-based replication because it produces the same change data results with fewer updates actually replicated. This type of replication reduces network load and can increase the availability of the target table.

You can implement transaction-consistent replication by using the internal CCD table model. In this model, outbound queues are condensed before replication, keeping only the latest captured value for each row. The Apply program condenses the queues by copying noncondensed change data already in the CD table into an internal (local) condensed CCD table. This CCD is then used as the source for replicating changes to the target table. Figure 9 on page 28 shows the configuration of an internal or local CCD table.

How External CCD Tables Are Refreshed (Cascade CCD Full Refresh)

When an Apply program refreshes an external CCD table, it deletes all of the rows from the pruning control table associated with the CCD table. The missing CCD table rows in the pruning control table indicate a full refresh and alert the Apply program that replication subscriptions based on the external CCD table must also be refreshed.

The Apply program keeps track of replication subscriptions based on the external CCD table in the following way:

- If the refreshed CCD table contains rows, the Apply program updates the value in the CCD_OLD_SYNCHPOINT column of the corresponding CCD row in the CD control table. The value is set to the minimum commit sequence value of the refresh rows applied to the CCD table.
- If the CCD table is empty as a result of the refresh, the CCD_OLD_SYNCHPOINT value is set to binary zeroes. You maintain the CCD_OLD_SYNCHPOINT value after pruning the CCD table because you are responsible for pruning the CCD table.

Developing a Data Warehouse with CCD Tables

You can define any number of replication subscriptions for CCD tables that refer to a source table. You can also define replication subscriptions referring to CCD tables that are remotely located from the original source table. In this sense, remote CCD tables become surrogate source tables. This introduces distribution databases or *warehouses* that serve as sources for all other copies. Changes captured on your operational systems need only be replicated once to the warehouse database to be applied to CCD tables in the warehouse database. These CCD staging tables can then serve as surrogate source tables for all other replication subscription definitions.

How the Apply Program Selects a Source Table

When the Apply program refreshes or updates a target, it chooses from a list of potential source tables in the following order:

1. The CCD table associated with the defined replication source table
2. The CD table associated with the replication source table
3. The replication source table associated with the replication subscription

In some cases, the source table considered first might not be the one that the Apply program reads from. For example, if the target table is a point-in-time table but has not yet been initialized, the Apply program must use a source table that is complete. If the CCD table is not complete (that is, CCD_COMPLETE=N in the register control table), the replication source table is selected as the refresh source. (The complete attributes are set when the CCD table is created.)

Using Changed Data to Update Condensed Copies

CCD tables and point-in-time (PIT) tables are changed, rather than appended. The originally captured operation code in the IBMSNAP_OPERATION column and the sequence numbers, IBMSNAP_INTENTSEQ and IBMSNAP_COMMITSEQ, are copied into and out of CCD staging tables. For condensed CCD tables, only the latest values are kept for any given row. The copy operation in IBMSNAP_OPERATION is an insert, update, or delete. The codes are:

I	Insert
U	Update
D	Delete

Exceptions:

- The operation code is I, but a unique index constraint causes the insert to fail because a row with the key already exists.

  Result: The insert becomes an update.
- The operation code is U, but the update fails because no row exists with the key.

  Result: The update becomes an insert.
- A delete fails with a “row not found” condition.

  Result: The condition is ignored by Apply.
- For a condensed CCD, a delete is always handled as an update.

Pull Versus Push Apply Design

The pull versus push configuration is a question of where the Apply program is running: at the source server or the target server. In the push method, the Apply program runs at the source server. In the pull method, the Apply program runs at the target server. The level of granularity is at the replication subscription level; one Apply program could be pushing for some replication subscriptions and pulling for others.

When the Apply program processes a replication subscription, it first connects to the source server to fetch the current changed data. This data is fetched into a spill file that is local to the Apply program. Once the data has been retrieved, the Apply program connects to the target server and applies the changes, one row at a time, as an INSERT, UPDATE, OR DELETE to each target table.

Figure 11 on page 33 shows the difference between push and pull modes.

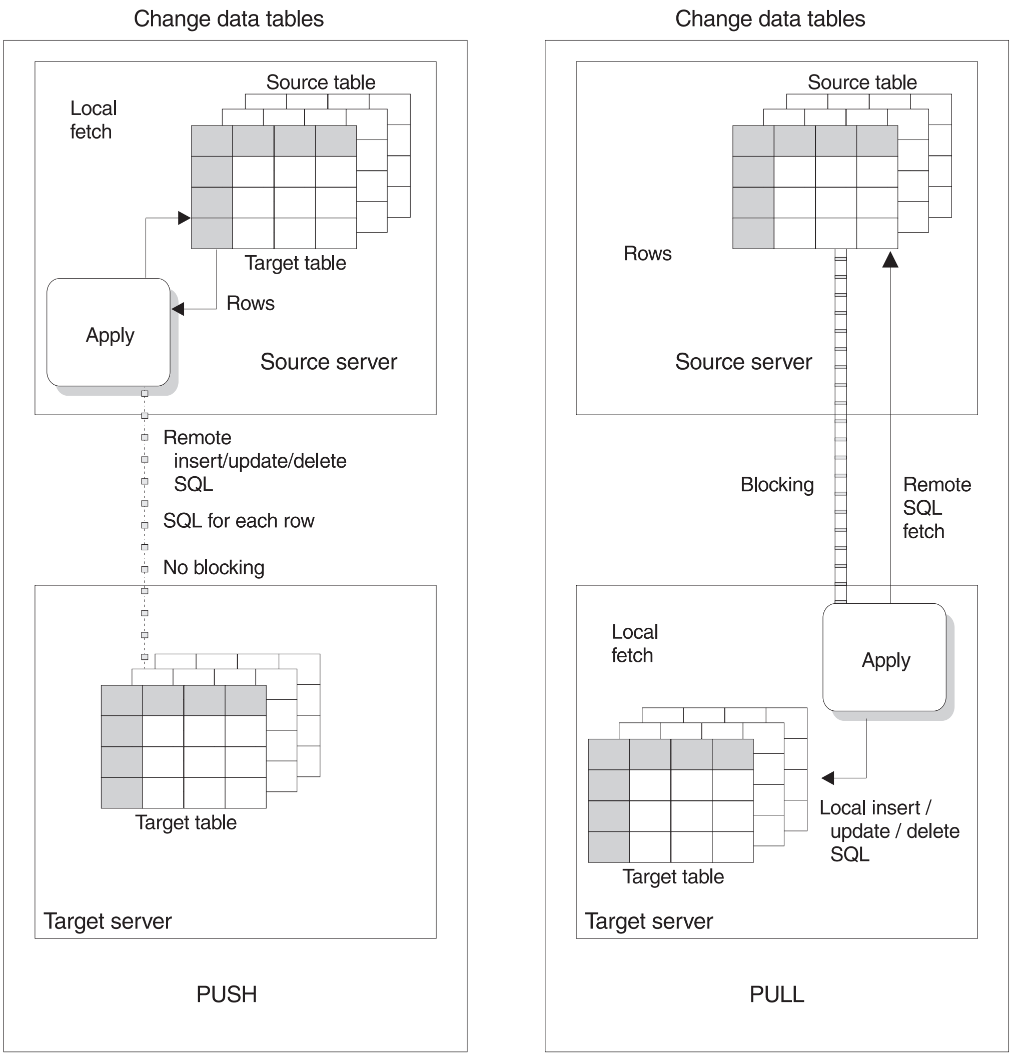


Figure 11. Push Versus Pull Mode

In pull mode, the Apply program connects to the remote source server to retrieve the data. DB2 can then use block fetch to pass the data across the network efficiently. When all data has been retrieved, the Apply program connects locally to the target server and applies the changes to the target table. The row-by-row process occurs as a local operation.

In push mode, the Apply program connects to the local source server and retrieves the data. Then it connects to the remote target server and pushes the updates to the target table. The row-by-row process occurs as a remote operation, with no blocking for network efficiency.

You do not have to do any special configuration to set up a push or pull configuration, except decide where to run the Apply program. The replication components and the Control Center recognize both configurations. The Control Center automatically sets up the replication control tables so the Apply program can push or pull data.

Generally, a pull configuration performs better than a push configuration because it allows more efficient use of the network. However, under the following circumstances a push configuration is a better choice:

- When there is no Apply program for the target server platform, for example, as with VSE or VM.
- The source table changes very infrequently, but when it changes it should be replicated as soon as possible.

Update-Anywhere Replication

Update-anywhere replication allows you to replicate changes from replication sources to target tables and from target tables back to the replication source table. With the Capture program running at both the source and target servers, changes against the original user table (the replication source object) and its updatable copy (replica) can both be captured, and the Apply program then replicates these changes from one server to the other to maintain data currency. This is also true for a one-to-many configuration (one source table to many replicas); changes to the replication source table can be copied to all of its replicas, and changes to any of the replicas can be copied to the replication source table. The changes are then copied to all other replicas.

Figure 12 on page 35 shows a basic update-anywhere replication model. Changes are originated at both the source and replica tables.

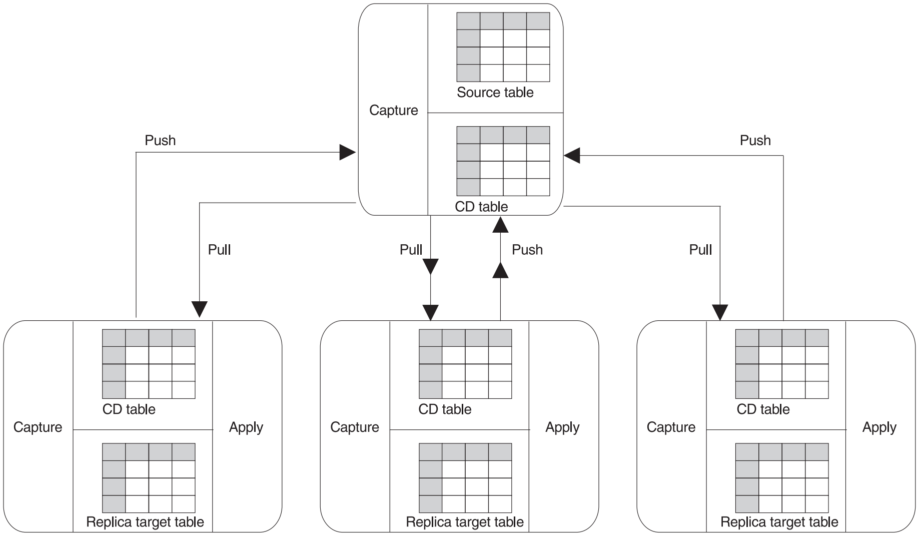


Figure 12. Update-Anywhere Replication. Update-anywhere replication is used to replicate changes from both the source and target servers.

The Apply program detects update conflicts, after they occur, during the subscription cycle. When transactions are rejected, the Apply program compensates the transactions at the replica. During this process, the Apply program inserts rejection codes for every rejected transaction in the IBMSNAP_REJ_CODE column of the UOW control table. Conflict detection is provided at three levels: no detection, standard detection, and enhanced detection, specified while defining the replication source. When a conflict is detected and compensated, the replication subscription fails. All related transactions are checked for conflicts and are rejected also.

Conflict detection cannot detect read-transaction conflicts. If, for example, an application reads information that is subsequently removed by compensation, the dependency cannot be detected.

After the Apply program completes the subscription cycle, the ASNDONE exit notifies you that the cycle is complete and whether there were conflicts. You can use the ASNDONE user exit to manage recovery from conflicts.

See “Update-Anywhere Replication” on page 76 for more detailed information about these concepts and suggestions for implementing update-anywhere replication. See the Capture and Apply program chapter for your platform in this book for more information about ASNDONE.

Views for Replication Sources and Targets

You can define source views or target subset views.

Source Views

Using the Control Center, you can define a join as a replication source. The joins can only include tables defined as replication sources. If the replication sources defined in the join have CD (or CCD) tables, a CD view is created from the replication sources' CD tables. The Capture program (or the Apply program, for CCD sources) maintains the control information for the joined replication sources and the CD views in the source server control tables.

Join views fill many requirements, both for denormalizing (restructuring) copies in data warehouse scenarios, enabling easier querying of copied data, and also for addressing the routing problem, sometimes called the database partitioning problem in distributed computing scenarios. For example, knowing where to send a bank account update may require a join of the account table with the customer table, in order to know which branch of the bank the customer deals with. Typically, production databases are normalized so that the geographic details, such as branch-number, are not stored redundantly throughout the production database.

IBM Replication supports the following types of view definitions:

- Simple inner-joins over one or more defined replication sources
- Simple inner-joins over CCD staging tables that are defined as replication sources and maintained by an Apply program or an application other than an IBM Replication component and an external data source, such as DataPropagator NonRelational and IMS source data

Target Views

You use the target views only when consolidating data from multiple sources, such as multisite union scenarios. Multiple source tables are updated and consolidated into a join at the target server. These types of views are not supported in the Control Center.

When the target table is maintained by different servers or different replication subscriptions, a subset view ensures that the DELETE statement generated before a full-refresh is copied to the appropriate horizontal fragment. The Apply program is almost always driven by SQL operations at the source server and in addition, generates an unqualified DELETE statement at the target server before applying a full-refresh answer set. By defining a subset view over the target table, and defining the replication subscription target to be the view, you restrict the DELETE statement to the appropriate horizontal fragment. That way, a full refresh of data from the source does not affect the information from the other sources.

Mobile Replication Enabler

The mobile replication enabler allows you to operate the Capture and Apply programs in mobile mode for the occasionally connected environment. For applications that support a mobile work force or one in which clients infrequently connect, the mobile mode allows you to replicate on demand, whenever the source or target server is online.

With the mobile replication enabler you can request immediate capture of changed data and that the Apply program replicate data immediately to reduce network costs. See Chapter 16, “Mobile Replication” on page 229 for planning, administration, and operating information.

How the Capture and Apply Programs Coordinate Pruning

The Capture program prunes the CD and UOW tables based on information inserted into the pruning control table by the Apply program. The Apply program maintains the log sequence number in the SYNCHPOINT column of the pruning control table. Initially, the Apply program sets this sequence number to zero when it performs a full refresh. A zero value signals the Capture program to start capturing. When the Apply program copies changes from the CD table to the target table, it updates the SYNCHPOINT column. The Capture program can then prune changes in the CD table up through the row with the highest log sequence number.

Pruning occurs depending on whether you start the Capture program with the PRUNE or NOPRUNE invocation parameter and how the prune interval is set in the tuning parameters table. See the Capture and Apply chapter for your platform in this book to learn how to set pruning.

The Capture program does not insert data into CCD tables and does not prune them. Instead, your application requirements should determine the history retention period for CCD tables (described in “Staging Changed Data” on page 24). Therefore, pruning of CCD tables is not automatic by default, but can be easily automated using an SQL statement to be processed after the subscription cycle.

Gaps Between Source and Target Tables

Occasionally, a gap can occur between the capturing of the changed data for a source table and the replication of the changed data to the target table. When this happens, a user with sufficient privileges might need to reset the control table information before executing the definition again. However, to preserve data integrity, check the control tables and take any necessary action.

For example, if you shut down the Capture program and then cold start it, it deletes all rows from the CD table. (See the Capture and Apply chapter for your platform in this book to learn more about cold start.) Between the time you shut down the Capture program and cold started it, updates might have been made that the Capture program did not capture. Additionally, any updates that were in the CD table were deleted at the cold start before the Apply program could copy them. A gap now exists between the target table and the CD table.

When a gap is present, the Apply program attempts to refresh complete copies unless the target table is not complete (COMPLETE=N). If the target table is not complete and the Apply program cannot perform a refresh, data integrity could be lost. In this case, before you reset the copy definition and resume copying, you need to check:

- The control information in the subscription set control table of the target table
- The target table itself
- The base table

The Capture and Apply programs use the synchpoint to coordinate their work. The synchpoint is the log sequence number and indicates the progress of the replication subscription through a subscription cycle. The Capture and Apply programs maintain and use this value to prevent the pruning of data that the Apply program did not copy. The synchpoint value is maintained in the SYNCHPOINT column of the subscription set control table.

To verify whether a gap exists:

Compare the SYNCHPOINT column in the subscription set control table with the CD_OLD_SYNCHPOINT column in the register control table. If SYNCHPOINT is lower than CD_OLD_SYNCHPOINT, a gap exists.

To resolve the gap problem:

Determine whether continued performance or data integrity is more important.

- If data integrity is important, manually insert the missing rows into the target table.
- If continued performance of the Apply program is more important, set CD_OLD_SYNCHPOINT to a lower value so that the Apply program continues regardless of the data loss.

Chapter 3. Getting Started

This chapter presents a high-level overview and practical application of the following steps required to set up a replication environment:

1. Plan for a replication scenario at the system level and application level as described in Chapter 5, “System Planning” on page 57 and Chapter 6, “Application Planning” on page 69.
2. Define a replication source as described in Chapter 8, “Working with Replication Sources” on page 93.
3. Define a replication subscription as described in Chapter 9, “Working with Replication Targets” on page 103.
4. Configure the Capture and Apply programs. See the Capture and Apply chapter for your particular platform for configuration information.
5. Start the Capture program as described in the Capture and Apply chapter for your particular platform.
6. Start the Apply program as described in the Capture and Apply chapter for your particular platform.

You do not need to refer to the detailed information referenced in these steps to perform the scenario in this chapter. The terms you will encounter, such as Apply qualifier, are explained in Chapter 2, “Introduction to IBM Replication: Tools and Concepts” on page 7 and in the glossary.

Performing a Replication Scenario

The hypothetical scenario in this section and the following sections enables you to acquire preliminary experience using the Control Center and the Capture and Apply programs. This scenario walks you through the steps above using hypothetical data to copy a simple DB2 source table to a target DB2 database for Windows NT. This type of copy is known as a *user copy*, which is a complete, condensed copy of the replication source table without the timestamp columns.

This scenario uses the sample source table shown in Table 4 on page 40. This is the DEPARTMENT table in the SAMPLE database that is contained in the DB2 Universal Database. You can create the SAMPLE database from the First Steps icon in the DB2 Universal Database.

Table 4. Sample Source Table Rows

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
A00	Spiffy Computer Service	000010	A00
B01	Planning	000020	A00
C01	Information Center	000030	A00
D01	Development Center	-	A00
D11	Manufacturing Systems	000060	D01
D21	Administration Systems	000070	D01
E01	Support Services	000050	A00
E11	Operations	000090	E01
E21	Software Support	000100	E01

Nomenclature for the scenario is as follows:

DEPARTMENT Source table name

DEPARTCOPY Target table name

SAMPLE Source server database

COPYDB Target server database

COPYDB Control server database

DEPTQUAL Apply qualifier

DB2 Instance name

This scenario accomplishes the following objectives:

- Prevents capture changes for the LOCATION column
- Prevents before-image changes for ADMRDEPT
- Captures changes for all other columns
- Sends copies to the target table at a rate of 1 copy each minute, ending only when Apply for NT is stopped

To set up the environment to run this scenario for Windows NT, do the following:

1. Create the SAMPLE database to be used as the source server database.
2. Create a database called COPYDB to be used as the target and control server database.
3. Verify that these databases appear in the Control Center.

The following sections explain the steps required to define a user copy for DEPARTMENT:

- “Defining a Replication Source” on page 41
- “Defining a Subscription” on page 42
- “Configuring the Capture and Apply Programs” on page 43
- “Starting the Capture and Apply Programs” on page 45

Defining a Replication Source

To define a replication source:

1. Open the Define as Replication Source window:
   a. From the Control Center, expand the object tree under the Systems icon until you see the SAMPLE database.
   b. Click on SAMPLE, then click on the Tables folder. Any existing tables are displayed in the pane on the right side of the Control Center.
   c. Click mouse button 2 on the source table, DEPARTMENT, and select **Define as Replication Source - Custom** from the pop-up menu.
2. From the **Available columns** box, select LOCATION and clear the **Define as source** check box. This specifies that this column is unavailable as a source column during subscription and cannot be replicated. This action also clears the **Capture before image** check box.
3. From the **Available columns** box, select ADMRDEPT and clear the **Capture before image** check box. This action prevents before image replication of the source column to the target table.
4. Click on **OK** to save the values and close the window. A confirmation window opens.
   a. Select the **Save to SQL file and run later** radio button and click on **OK**. The Control Center will generate the necessary SQL.
   b. Type the file name where you want to save the file in the Run SQL files window and click on **OK**.
   c. Click on **OK** in the DB2 message window to save the values and close the Define as Replication Source window.
5. Click mouse button 2 on the **Replication Sources** folder and select **Run SQL files**.
6. Run the SQL file you named above.
7. Click on **OK** in the DB2 message window when the message appears that the script file has run successfully.
8. Click mouse button 1 on the **Replication Sources** folder. The replication source, DEPARTMENT, appears in the contents pane of the Control Center.

Now the source table DEPARTMENT has been defined as a replication sources table. Additionally, the change data table for DEPARTMENT and all the Capture control tables are created in the default table space for the SAMPLE database. If you want to create the tables in other table spaces, see “Working with Customized Replication Control Tables” on page 88 and “Customizing and Running Replication SQL Files” on page 89 for more information.

Defining a Subscription

To define a subscription to copy the changes from DEPARTMENT, the source table, to the target table named DEPARTCOPY:

1. From the Replication Sources list displayed in 8 on page 41 above, click mouse button 2 on the DEPARTMENT object on the right pane of the Control Center, then select **Define subscription**. The Define Subscription window opens.
2. Set up the target table and subscription definition.
   a. Type `DEPTSUB` in the **Subscription name** field.
   b. Type `COPYDB` in the **Target server** field. This database is where the target table will reside.
   c. Type `DEPTQUAL` in the **Apply qualifier** field. The character string identifies the subscription definitions unique to each instance of the Apply program that will run this subscription.
   d. Type `DEPARTCOPY` over the default name of the target table. The Control Center adds a table qualifier prefix to the target table name.
   e. Specify that you want to create the target table by selecting the **Create table** check box for the `DEPARTCOPY` target table.
3. Click on the **Advanced** push button. The Target Type page of the Advanced Subscription Definition notebook opens.
   a. Select the **User Copy** radio button.
   b. Click on the **Target Columns** tab.
      1) Select the check box under **Subscribe** for XDEPTNO.
      2) Check the **Primary key** check box next to DEPTNO.
   c. Click on the **Rows** tab and type the following WHERE clause:

      ```
      DEPTNO >='A00'
      ```

   d. Click on **OK** to save these settings and return to the Define Subscription window.
4. Define SQL statements to be processed during subscription run time:
   a. Click on the **SQL** push button. The SQL window opens.
   b. Click on the **Add** push button. The Add SQL window opens.
   c. Type the following processing statement in the **SQL statement** field:

      ```
      DELETE FROM ASN.IBMSNAP_APPLYTRAIL
      WHERE LASTRUN < (CURRENT TIMESTAMP - 1 DAY)
      ```

   d. Type the SQLSTATE value `02000` in the **SQLSTATE** field and click on **Add**. This value is added to the **Acceptable SQLSTATE values** list box.
   e. Click on the radio button that indicates you want to submit the statement before the subscription is processed.

   f. Click on **OK**. The statement is added to the list box in the SQL window and the Add SQL window closes.

   g. Click on **OK** to return to the Define Subscription window.

5. Specify when and how often to replicate the subscription by clicking on the **Timing** push button. The Source to Target page of the Subscription Timing notebook opens.

   a. Keep the current date in the **Start date** field.

   b. Keep the current time in the **Start time** field.

   c. Select the **Time-based** check box and the **Using relative timing** radio button.

   d. Use the spin buttons on the **Minutes** field to select 1-minute intervals.

   e. Use the spin buttons on the **Hours** field to change the default number to 0.

   f. Click on the **Data Blocking** tab.

   g. Click on the spin buttons and select 5 as the number of minutes at a time that Apply will copy committed data.

   h. Click on **OK** to save these values and return to the Define Subscription window.

6. Click on **OK** to submit the subscription. The Subscription Information window opens.

   a. Type in `COPYDB`.

   b. Select the **Save to SQL file and run later** radio button and click on **OK**.

   c. Type the file name where you want to save the file in the Run SQL files window and click on **OK**.

   d. Click on **OK** in the DB2 message window to save the values and close the Define Subscription window.

7. Click mouse button 2 on the **Replication Subscriptions** folder under the SAMPLE database and select **Run SQL files**.

   a. Run the SQL file you named above.

   b. Click on **OK** in the DB2 message window when the message appears that this file has run successfully.

8. Click mouse button 1 on the **Replication Subscriptions** folder under the SAMPLE database. The DEPTSUB subscription appears as an instance on the contents pane of the Control Center.

Configuring the Capture and Apply Programs

The following sections provide instructions for configuring the Capture and Apply programs for Windows NT, and for providing end-user authentication at the source server.

Configuring the Capture Program

To configure the Capture program:

1. From the source server system, ensure that the user ID under which the Capture program is to run has the required privileges:
   - Execute privilege on the Capture packages
   - DBADM or SYSADM privileges for the database
2. Log on with the user ID.
3. From the DB2 command line processor or the Command Center, connect to the source server database by entering:

   ```
   DB2 CONNECT TO SAMPLE
   ```
4. Change to the \SQLLIB\BND directory, which is where the bind files are located.
5. Create and bind the Capture program package to the source server database by entering the following command:

   ```
   DB2 BIND @CAPTURE.LST ISOLATION UR BLOCKING ALL
   ```

   If your system does not support uncommitted read format, substitute `CS` (cursor stability format) for `UR` instead.

   These commands create a list of packages, the names of which can be found in the CAPTURE.LST file.

Configuring the Apply Program

To configure the Apply program:

1. From the source, control, and target server system, ensure that the user ID under which the Apply program is to run has the required privileges:
   - Execute privilege on ASNAPPLY
   - DBADM or SYSADM privileges for the database
2. Log on with the user ID.
3. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO SAMPLE
   ```
4. Create and bind the Apply package to the source server database by entering both of the following commands:

   ```
   DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL
   ```

   ```
   DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL
   ```

   These commands create a list of packages, the names of which can be found in the APPLYUR.LST and APPLYCS.LST files.
5. Connect to the target server database by entering:

   ```
   DB2 CONNECT TO COPYDB
   ```
6. Create and bind the Apply package to the target server database by entering both of the following commands:

```
DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL
```

```
DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL
```

These commands create a list of packages, the names of which can be found in the APPLYUR.LST and APPLYCS.LST files.

Providing End-User Authentication at the Source Server

For end-user authentication to occur at the source server, you must provide a password file with an AUTH=SERVER scheme for the Apply program to use when connecting to the source server. It is advisable to limit read access of this file to the user ID that will run the Apply program.

The password file must meet the following criteria:

- Have a file name as shown:

  `<APPLYQUAL><instname><CNTLSRVR>.PWD`

  Where:

  APPLYQUAL
  The Apply qualifier in uppercase.

  instname
  The instance name that the Apply program runs under (value of DB2INSTANCE).

  CNTLSRVR
  The name of the control server in uppercase.

- Reside in the directory from which the Apply program starts.
- Contain all server-name/password pairs. The pairs enable you to have a different (or the same) password at each control server. The Apply user ID is used for all connections.

To create a password file:

1. Type the following records in your editor. Do not include blank lines or comment lines in the file.

   ```
   SERVER=SAMPLE PWD=password
   ```

   ```
   SERVER=COPYDB PWD=password
   ```

2. Save the file as DEPTQUALdb2COPYDB.PWD.

For more information about authentication and security, refer to the *IBM DB2 Administration Guide for common servers.*

Starting the Capture and Apply Programs

After defining the source and subscription, you can submit the copy request by starting the Capture and Apply programs.

To start the Capture program from the SAMPLE source server:

1. Enter the following command from the Windows NT window:

   ```
   ASNCCP SAMPLE
   ```

   No confirmation message appears. The Capture program has successfully started if it is running and no command prompt appears. See the troubleshooting section in your particular platform chapter for problem determination.

To start the Apply program:

1. Enter the following command from another Windows NT window:

   ```
   ASNAPPLY DEPTQUAL COPYDB
   ```

   No confirmation message appears. The Apply program has successfully started if the program is running and no command prompt appears. See the troubleshooting section in your particular platform chapter for problem determination.

Stopping the Capture and Apply Programs

To stop the Capture and Apply programs after the user copy replication is complete:

1. Stop the Capture program.

   a. From a DB2 command window, use the SET command to set the environment variable DB2DBDFT to the source server name:

      ```
      SET DB2DBDFT=SAMPLE
      ```

   b. Enter the following command from the same DB2 command window:

      ```
      ASNCMD STOP
      ```

   You can alternatively use one of the following key combinations from the window where the Capture program is running:

   - Ctrl+C
   - Ctrl+Break

2. Stop the Apply program.

   a. From a DB2 command window, use the SET command to set the environment variable DB2DBDFT to the target server name:

      ```
      SET DB2DBDFT=COPYDB
      ```

   b. Enter the following command from the same DB2 command window:

      ```
      ASNASTOP DEPTQUAL
      ```

   You can alternatively use one of the following key combinations from the window where the Apply program is running:

   - Ctrl+C
   - Ctrl+Break

Sample data for the target table results after one iteration are shown in Table 5 on page 47. Note that the order of the target table columns has changed, but the data remains the same as for the sample source table.

Table 5. DEPARTCOPY Table Results

ADMRDEPT	DEPTNAME	XDEPTNO	DEPTNO	MGRNO
A00	Spiffy Computer Service	—	A00	000010
A00	Planning	—	B01	000020
A00	Information Center	—	C01	000030
A00	Development Center	—	D01	-
D01	Manufacturing Systems	—	D11	000060
D01	Administration Systems	—	D21	000070
A00	Support Services	—	E01	000050
E01	Operations	—	E11	000090
E01	Software Support	—	E21	000100

Part 2. Planning for Replication

This part describes the planning issues you must consider before setting up your replication environment. It discusses recommended replication scenarios that have been implemented by other customers and planning issues at the system and application level.

Chapter 4. Replication Usage Scenarios

This chapter describes customer environments where IBM Replication has been proven to work successfully, some rather more complex environments where the product can be used with some caution, and environments that are not supported. The scenarios covered here are not exhaustive because customers are always coming up with new and creative ways of using IBM Replication. The key is to know in advance what your chances of success are!

Recommended Usage Scenarios

Unless otherwise indicated, each of the scenarios in this section is a tried and tested implementation based on actual customer installations. [2]

Operational to Decision Support System Data Replication

One of the common uses of IBM Replication is to replicate data changes from an operational system to a decision support system. Generally it is not advisable to run decision support and online systems alongside each other. A process is therefore required to maintain the decision support system in line with the operational system.

IBM Replication solved this problem for one financial institution that had to replicate updates from its customer information database held on DB2 for MVS to a decision support system that was also to be on DB2 for MVS. The main requirements were that the operational system not require any code modifications and the performance of the operational system not be impacted in any way.

IBM Replication was implemented to capture updates from the key operational tables and, on an hourly basis, replicate them to consistent change data (CCD) tables in the decision support system (DSS) DB2 subsystem. CCD tables were used because history information was required in the DSS, but none was maintained in the operational system. The IBM Replication tasks were given an MVS dispatching priority such that no CPU resource was ever taken away from the operational system. If there was a shortfall in capacity during peak periods, the Capture program task simply lagged behind.

The decision support system could be implemented just as easily on any of the supported target platforms and could still be ported to other platforms at some time in the future if required.

[2] In many cases some of the details have been changed to conceal the real identity of the customer. Sometimes two customer scenarios have been combined.

Distributed Database Systems

A large European retail chain has almost 500 stores around the country, each of which gathers purchase details through an electronic point of sale (EPOS) system. The data is transferred nightly to a central DB2 for MVS site using a preexisting file transfer process from the EPOS terminals. The company wants to leverage the value of the data by enhancing it at the central site and then distributing it to each of the branches and to some regional offices.

IBM Replication solved the problem by providing a weekly, controlled distribution of multilevel summarized results. Apply for AIX is used to summarize the details from the MVS site before transporting them and applying them to the local DB2 for AIX databases. Each store is able to see the local trends and their performance against other outlets in the region. Regional managers are able to more effectively plan their marketing and distribution strategies by using the information to help set objectives for the individual stores.

Improved Network Load Balancing

A small bank rolled out several new OS/2-based client/server applications to its 85 branches. A major source of data for the new applications is the customer and financial reference data, which is derived and held in two operational systems, one on DB2 for MVS and the other on DB2 for AIX. It was not possible to directly access the host site for all data requirements, so a data replication solution that uses IBM Replication was devised.

Changes arising from the two operational systems throughout the day are captured and staged on the AIX platform. Several CCD tables are built, with Apply for AIX taking feeds from both DB2 for MVS and the local DB2 for AIX system. Overnight, the changes are Replicated to the OS/2 applications in the branches, ready for use the next morning.

This approach helps to minimize the network traffic by maintaining a local copy of the database in each branch. Furthermore, IBM Replication replicates*only* the changed customer and financial data and ensures that only the most recent update for each key is replicated, thus reducing network requirements.

Data Consolidation and Distribution

A large financial institution wanted to improve the flow of information from two legacy operational systems to its OS/2-based branch organization. The intent was to provide more accurate and timely data to help loan application research and to detect credit card fraud. The data to drive the loan application work was in DB2 for MVS, and the credit card details were captured in an older IMS-based system. Previous attempts to copy the legacy data had been an unworkable mixture of ad hoc reports and file transfer techniques.

Today, using IBM Replication, the branch analysts enjoy a much improved service, with daily feeds from the IMS system and twice weekly downloads of the loan history details. The DB2 for MVS-based data is captured daily but staged in a CCD table for the two weekly replication jobs. Only data that has changed is replicated, and the changes are replicated to provide a historical perspective. The credit card information is captured from IMS with DataPropagator NonRelational. The resulting staging tables are then replicated to the branches.

Improved Application Availability

An international bank has a very sensitive online environment. The online system is used 23 hours 45 minutes a day. Every day the bank must stop the system to quiesce it for a batch application. The batch application requires exactly one day's worth of data. This data is needed for a day account. During the 15 minutes when the system is down, the bank extracts the tables required. When the bank finishes the extraction, the online system is made available for the next financial day.

To meet its goal of using the online system 24 hours a day, but not lose its application quiesce point, the bank decided to use IBM Replication. Data changes made during the online day were captured and then replicated to CCD staging tables. The batch application was modified to take input from CCD tables rather than extract files. The quiesce point can be identified from the data captured.

The source server and copy server are the same DB2 subsystem, so no distributed access is required except for the Control Center.

Data Archive

The archiving of aged or redundant DB2 data has long been a challenge for many large organizations. With IBM Replication, aged or redundant data can be safely deleted from the main tables without worrying about the archive database. The tables that require archiving are defined as replication sources. In doing this, whenever data is deleted, IBM Replication captures the deleted data and can “resurrect” the deleted row in a new target CCD table. The CCD table becomes the archive database.

Building Audit Trails

IBM Replication can be used to track changes to relational tables for auditing purposes, to determine which users made particular changes to the data.

One customer installation, for example, generated audit data by writing audit information to the IMS log, which was not a problem while all database access was through IMS. However, new applications were developed that accessed DB2 through DRDA, bypassing IMS completely. In this instance, IBM Replication was implemented to capture the updates. CCD tables were used, requesting both before and after image columns. The IBMSNAP_AUTHTKN column was replicated from the unit-of-work (UOW) control table to identify who performed the update.

Mobile Replication

Because of increasing competition from telephone-based direct insurance providers, an established insurance company wants to be more focused and aggressive in its sales campaigns. It wants to equip its sales representatives and agents, who rarely visit the company's office, with a set of offers to attract both new and existing customers—special introductory offers, personalized packages, special “today-only” offers, and tailored cross-selling offers.

The sales force is supplied with laptop computers running OS/2 and DB2 for OS/2. As a sales campaign is launched, custom programming helps to target customers, and IBM Replication is used to download the targeted customer profiles and history, as well as the latest product offers. IBM Replication also solves the problem of keeping the information up to date with regular dial-in downloads. Of course, only new and changed data rows are copied across the network.

See “Mobile Replication Requirements” on page 72 for more planning information.

Potential Usage Scenarios

This section is an unusual but important one: it tells you where IBM Replication can be used, but where it is recommended that care be taken in designing the system. The scenarios described in this section are complicated and difficult to implement as first replication projects.

Update-Anywhere Replication

IBM Replication supports update-anywhere replication, in which multiple copies (or replicas) of the same source table are synchronized and each replica can be a source of updates to the source table and the other replicas. This scenario allows you to set up circular subscriptions, where Table A updates Table B and Table B updates Table A. Customers can have a global database that updates and is updated by multiple replica target tables at multiple sites, such as bank branches or a mobile sales force.

Update-anywhere replication scenarios work best when transaction conflicts between the global database and the replicas can be avoided, such as when replicas can update only key ranges at specific sites, or when sites can make updates during certain time periods.

Conflict detection is provided at three levels and can be a data integrity and performance issue that should be well planned. See “Update-Anywhere Replication” on page 76 for more planning information.

Logical Recovery

IBM Replication can capture both before and after images of any update captured and the correlation ID (if using Capture for MVS) or the authorization ID of the user that performed the update. Using this information, it is possible to write some code that can “undo” updates that have been made to a given table. This approach could prove useful in the event of a rogue batch application incorrectly updating data. You may want to undo its updates but not the updates of any other application. However, care is required as the integrity of the database could be jeopardized if the application processing is not fully understood.

Extending IBM Replication

There are occasions when the data transformation required between source and target tables is so complex that IBM Replication cannot provide an end-to-end solution. In general, if the manipulation can be achieved with SQL, then IBM Replication can perform the translation for you. The use of SQL extends to procedural logic on some database platforms. DB2 Universal Database triggers and user-defined functions and triggers can be a part of the SQL invoked to perform the transformations.

Alternatively, it is reasonably easy to develop user code to read CCD tables populated by IBM Replication, perform any transformations required, and then apply the changes to the target tables. See “Defining SQL Statements or CALL Procedures for the Replication Subscription” on page 116 for more information about using stored procedures before or after the changes are applied to the target tables.

Usage Scenarios Not Recommended for Replication

This section describes scenarios that IBM Replication was not designed to support. In these cases, you must look for other solutions. Hints are given as to which products or techniques may be helpful.

Synchronous Replication

IBM Replication does not support synchronous replication. However, lack of support is often the benefit of using IBM Replication as the base application is not affected. If the network connection from the base application to the target is unavailable, the base system is still be available if you use IBM Replication, and data changes are not lost.

In many cases, organizations that start out with an initial requirement for immediate availability of copy data find that the benefits of staged delivery (such as, better network use, less database contention, and the opportunity to enhance the data) are more attractive, so they reexamine the original requirements. IBM Replication supports continuous timing of subscriptions.

If synchronous data delivery is essential to the application, the recommended approach is to use DRDA two-phase commit within the application.

Hot-Site Recovery

Many planners are tempted to consider IBM Replication for hot site recovery because of its ability to replicate tables to other sites. However, IBM Replication was not designed for such recovery because the underlying architecture is asynchronous. There is no guarantee of how much data has been captured or applied at any one instant.

Hot site recovery is a complex and specialized task and is supported by several IBM and vendor features and tools. For example, in the System/390 area, the peer-to-peer remote copy (PPRC) hardware feature and the extended recovery component (XRC) are available.

Chapter 5. System Planning

This chapter helps to establish the feasibility of the replication implementation by considering the system requirements: which replication products to use, software and hardware requirements, storage requirements, CPU capacity, network arrangement, security authorizations, and user IDs.

Even if your requirements are not fully defined at this stage, this chapter will start you thinking about the kinds of questions that you should be asking.

Replication Products

IBM Replication products come packaged either in product solutions or are available separately. Table 6 describes how the administration tools and Capture and Apply programs are packaged.

Table 6 (Page 1 of 2). The Replication products packaging solution

Product Name	Platforms	Replication Features	Includes Control Center?	Includes DB2 Connect?	Includes DB2?	Includes Client Application Enabler (CAE)?
DB2 Enterprise Edition	OS/2, Windows NT, AIX, HP/UX, Solaris	Capture program, Apply program, and DPROPR V1 Migration program on OS/2, Windows NT, Windows 95	Yes, for OS/2, Windows NT, and Windows 95. For UNIX, see (1)	Yes, multi-user	Yes	Yes
DB2 Work Group Edition	OS/2, Windows NT, AIX, HP/UX, Solaris	Capture program, Apply program, and DPROPR V1 Migration program on OS/2, Windows NT, Windows 95	Yes, for OS/2, Windows NT, Windows 95. For UNIX, see (1)	No	Yes	Yes
DB2 Personal Edition(2)	OS/2, Windows NT, Windows 95	Capture program, Apply program, and DPROPR V1 Migration program on OS/2, Windows NT, Windows 95	Yes	No	Yes: single user	No
DB2 Connect Personal Edition(3)	OS/2, Windows NT, Windows 95, Windows 3.1	DPROPR V1 Migration program on OS/2, Windows NT, Windows 95	Yes, OS/2, Windows NT, Windows 95	Yes, single-user; DRDA-AR support; integrated SNA support	No	No

Table 6 (Page 2 of 2). The Replication products packaging solution

Product Name	Platforms	Replication Features	Includes Control Center?	Includes DB2 Connect?	Includes DB2?	Includes Client Application Enabler (CAE)?
DB2 Connect Enterprise Edition(2)	OS/2, Windows NT, AIX, HP/UX, Solaris	DPROPR V1 Migration program on OS/2, Windows NT, Windows 95	Yes, for OS/2, Windows NT, Windows 95. For UNIX, see (1)	Yes, multi-user	No	Yes
DPROPR Capture for MVS, V5.1	MVS	Capture program	Yes, packaged with DB2 Connect Personal Edition	Yes, packaged with DB2 Connect Personal Edition	No	No
DPROPR Apply for MVS, V5.1	MVS	Apply program	Yes, packaged with DB2 Connect Personal Edition	Yes, packaged with DB2 Connect Personal Edition	No	No
DB2 Server for VSE, 5.1	VSE	Capture Program	Yes, packaged with DB2 Connect Personal Edition	Yes, packaged with DB2 Connect Personal Edition	Yes	No
DB2 Server for VM, 5.1	VM	Capture program	Yes, packaged with DB2 Connect Personal Edition	Yes, packaged with DB2 Connect Personal Edition	Yes	No

(1) For UNIX platforms, the Control Center is also available in the Client Application Enablers for OS/2, Windows NT, and Windows 95, provided in the program package.

(2) Both products are required at a minimum for the mobile environment.

(3) Included in the Capture and Apply program packages for MVS, and Capture program packages for VSE and VM

Software and Hardware Requirements

The following section describes the hardware and software requirements for the replication tools.

The Capture and Apply Programs on the DB2 Universal Database

See the “Server Product Requirements” under “Hardware Requirements” and “Software Requirements” in *DB2 Universal Database Quick Beginnings* for your platform.

The Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM

The following sections describe the hardware and software requirements for the Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM.

Hardware Requirements

Table 7 on page 59 lists the hardware requirements for the Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM.

Table 7. Summary of Hardware Requirements for the Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM

Product	Requirements
Capture for MVS, 5.1	Any hardware that supports MVS/ESA Version 5 Release 1 or higher (for data sharing) System/390 parallel sysplex is required for data sharing
Apply for MVS, 5.1	Any hardware that supports MVS/ESA Version 5 Release 1 or higher (for data sharing) System/390 parallel sysplex is required for data sharing
Capture for VSE, 5.1	Any hardware that supports VSE/ESA Version 2 Release 1 Modification 2 or higher
Capture for VM, 5.1	Any hardware that supports VM/ESA Version 2 Release 1 or higher

Software Requirements

Table 8 lists the software requirements for the Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM.

Table 8. Summary of Software Requirements for the Capture and Apply Programs for DB2 for MVS, DB2 for VSE, or DB2 for VM

Product	Requirements
Capture for MVS, 5.1	• IBM MVS/ESA Version 4 Release 3 (5695-047) or higher (data sharing requires MVS/ESA Version 5 or higher) • DFSMS Version 1 Release 1 (5695-DF1) or MVS/DFP Version 3 Release 3 (5665-XA3) or higher • IBM C/370 Library Version 2 or IBM SAA AD/Cycle Language Environment/370 Version 1 Release 3 or higher, including any PTFs • IBM DATABASE 2 Server for OS/390, Version 5, Release 1 (5648-158), or IBM DATABASE 2 for MVS Version 3 Release 1 or higher, including any PTFs • IBM System Modification Program/Extended Version 1 Release 8 (5668-949)
Apply for MVS, 5.1	• IBM MVS/ESA Version 4 Release 3 (5695-DF1) or higher (data sharing requires MVS/ESA Version 5 or higher) • DFSMS Version 1 Release 1 (5695-DF1) or MVS/DFP Version 3 Release 3 (5665-XA3) or higher • IBM C/370 Library Version 2 or IBM SAA AD/Cycle Language Environment/370 Version 1 Release 3 or higher, including any PTFs • IBM DATABASE 2 Server for OS/390, Version 5, Release 1 (5648-158), or IBM DATABASE 2 for MVS Version 3 Release 1 or higher, including any PTFs • IBM System Modification Program/Extended Version 1 Release 8 (5668-949)
Capture for VSE, 5.1	• IBM DATABASE 2 Server for VSE & VM Version 5, Release 1 (5648-158) • IBM C/370 Run-Time Library Version 2, Release 1 (5688-188)
Capture for VM, 5.1	• IBM DATABASE 2 Server for VSE & VM Version 5, Release 1 (5648-158) • IBM Language Environment Version 1 Release 5 (5688-198) or higher, or the equivelent C & CEL run-time support provided in the base of VM/ESA Version 2

Storage Requirements

The additional storage requirements associated with the introduction of replication into your system will arise in three areas:

1. **Database management system log and journal data** — the additional data logged to support the capturing and applying of data.
2. **New tables** — all of the copied user data and any IBM Replication control tables, including staging tables.
3. **The Apply spill file** — the Apply program requires temporary space to store refresh answer sets. (With Apply for MVS, the spill file function can use memory rather than disk.)

DBMS Logging

If update copying is required, the database management system (DBMS) must log full row images for any update made to a table that has been selected for propagation. Additional log data is also generated when data is inserted or deleted from the unit-of-work (UOW) and change data (CD) tables in DB2.

It is not a straightforward calculation to estimate the increase that will be experienced in the log or journal volume. As a rule of thumb, it is recommended that the estimated increase be sized at three times the current log volume for the tables subject to replication.

A more detailed estimate requires detailed knowledge of the updating application and the replication requirements. If an updating application typically updates 60% of the columns in a table, the DATA CAPTURE CHANGES requirement will cause these log records to grow by more than half. Log records that would typically hold more or less of the row data would grow less or more, respectively. Several customers report a 60%-80% increase in the size of the update log records for the application tables that will be captured.

If only a small number of the columns are defined in the replication source, the inserts and deletes that are made to the CD table will cause small log records to be generated for these updates, whereas a replication source with all before and after images of the columns of a table will generate a larger log volume, as in the case of update-anywhere scenarios.

There will also be logging on the target database, where the rows are applied. Because the Apply program does not issue interim checkpoints, an estimate should be made for the maximum answer set of changed rows that will be handled by the Apply program, and the space for the logs should be allocated accordingly.

New Tables

The estimates for this storage are also based on the source updating applications and the replication requirements. The tables that account for the most storage use in replication are the UOW table, the CD tables, and the target tables. Space requirements can be estimated for the target tables based on the source tables.

Space requirements for the CD tables can be estimated by first determining the “holding” period of the data. For example, if the Apply program is run once daily, with a “pruning” deletion of the applied rows from the CD table occurring once a day just after the Apply program, the base "holding" period would be 24 hours worth of updates. See “How the Change Data and UOW Tables are Pruned” on page 13. If the CD rows are 100 bytes long, and 100,000 updates are captured on an average day, the storage required for the CD table is about 10 MB. This is a sufficient approximation for initial storage requirement estimates. The CD table rows include 21 bytes of overhead data, in addition to the set of columns defined in the replication source.

The UOW table size is harder to estimate. The number of rows inserted in a day would equal the number of commits issued on transactions that update tables subject to replication. You should initially overestimate the size required and monitor the space actually used to see whether any space can be recovered. UOW table rows are fixed at 79 bytes long.

The space requirements for the IBM Replication control tables other than the UOW and CD tables are generally quite small. Most control tables require only a few rows, and default sizings are usually adequate.

The target table can be estimated by starting with the size of the source table, and then adding or subtracting changes to the target table from data transformation (adding columns), aggregating columns, or subsetting columns or rows.

The Apply Spill File

The spill file used by the Apply program is equal to the size of the data selected for replication at each interval, including full refresh.[3] The spill file uses disk storage on every platform, or it can use virtual memory with Apply for MVS. The size of the answer set to be selected by the Apply program can be estimated by looking at the frequency interval planned for the Apply program, as compared to the volume of changes in that same time period, or in the peak period of change. For example, if change volume peaks at 12,000 updates per hour, and the Apply program frequency is planned at one-hour intervals, the one-hour interval will determine the average size of the spill file. In this case, the spill file must hold one-hour's worth of updates, or, 12,000 updates. The row size is the *target row* size, including any IBM Replication overhead columns. This row size will not be in DB2 packed internal format, but in expanded, interpreted character format, as fetched from the select. The row also includes a row length and null terminators on the individual column strings. Multiple spill files are used for replication subscriptions with multiple target tables; one spill file for each target table.

Data Blocking for Large Volumes of Changes

You may have a replication subscription with a large block of changes from one or more CD or CCD tables to be replicated in one Apply cycle. These large loads can cause contention between the Capture and Apply programs. They can also cause the spill file or log to overflow.

[3] If you are using ASNLOAD, instead of a load spill file there will be a load input file.

Use data blocking to break up the backlog of large batch volumes into mini-subscription cycles, using multiple real subscription cycles to perform one scheduled cycle. The Apply program breaks down the answer set into smaller blocks, preventing spill file and log overflows.

You specify the number of minutes of data that the Apply program copies at a time. The number of minutes you specify is used to determine the size of the data block. Use the **Data Blocking** page of the Subscription Timing notebook to indicate the number of minutes at a time that the Apply program will copy committed data.

The data blocking minute default is 0 (zero). The number of minutes that you set should be small enough so that all the transactions for the replication subscription that occurred during the interval can be copied without causing the spill file to overflow or a log full condition. The space required for the spill files is the sum of the space required for each answer set in the replication subscription.

The Apply program uses the user-inputted interval initially. If the value is too high, the Apply program must retry frequently using smaller interval values. It automatically retries three times, adapting to the work load level to copy in blocks that won't cause spill file or log overflow. To adapt the work load, the Apply program cuts the value in half to get the answer set size small enough to replicate successfully. If there is still a problem, it uses one-fourth of the value to align the changed data backlog with the available system resources.

Restrictions:

- You can't split a unit of work
- You can't roll back previous mini-subscription cycles
- You only use data blocking in update mode, not full refresh

The Active Log File Size for Capture for MVS on DB2 for MVS 3.1 and Capture for VSE and VM 5.1

The Capture program captures updates from the log file. When the log is full, its contents are archived on an archive log file. If the system handles a large number of transactions, the Capture program can occasionally lag behind. If the log is too small, some of the log records can be archived before they are captured. Archived log records cannot be recovered by Capture for MVS, VSE, and VM running with DB2 for MVS 3.1 or DB2 for VSE and VM 5.1. However, archived log records can be recovered by Capture for MVS running with DB2 for MVS 4.1 or higher, or the DB2 Universal Database V5.

If you are running on DB2 for MVS 3.1 or DB2 for VSE and VM 5.1, the size of your log should be large enough to handle at least 24 hours of transaction data.

General Storage Considerations

All of these sizes are estimates only. To prepare and design a production-ready system, factors such as failure prevention must also be taken into account. For example, the holding period of data might need to be increased to account for potential line outage.

If storage estimates seem to be unreasonably high, the frequency interval of the Apply program and pruning should be reexamined. Trade-offs frequently must be considered between storage utilization, capacity for failure tolerance, and CPU overhead consumed.

Capacity Requirements Analysis

IBM Replication requires processor resources in support of the Capture and Apply programs and the Control Center for replication administration.

The Capture Program

The CPU requirements for the Capture program are generally low. In a well-tuned environment, customers report that the Capture program does not impact the updating applications and requires a minimum of CPU capacity. Capture for MVS can be scheduled at a lower priority than the source applications. The Capture program will simply lag behind during periods where CPU resource is constrained.

The Capture program needs CPU resource during pruning, and this action can be deferred to times when system impact is low.

The Apply Program

The Apply program CPU requirements can vary greatly. The main factor that affects CPU is the currency requirement of the target system. The more frequent the propagation, the higher the overhead per propagated row and the greater the CPU consumption.

The Apply program requires CPU on both the source and the target system, where they are different. For example, Apply for OS/2 could be a Distributed Database Facility (DDF) thread performing mainly SQL select, but some SQL update activity, on a DB2 for MVS system. Estimating the cost of remote threads in a planned large distribution scenario might lead to a design involving an interim staging platform, as described in “Staging Changed Data” on page 24.

Administration

Replication administration can be planned for a time when impact to the source and target database systems will be minimal.

In general, the impact of the Control Center is not significant in terms of the local CPU. However, during times of intensive replication source and subscription definitions, the catalogs of the data server site are extensively searched. For large MVS sites, this can have a noticeable CPU or database system impact.

See the *Hardware Requirements* section in the *DB2 Universal Database Quick Beginnings* book for your platform to determine memory and disk requirements

Network Requirements

Anticipating the basic networking requirements for replication is easy—connect all of the sources to the targets, and connect the control point to all sources and targets for which it will perform administrative tasks.

Deciding among the various possible connectivity scenarios, estimating how much capacity will be required, and determining what level of data currency will be possible given the current available bandwidth can be very difficult chores, however. This section describes connectivity possibilities, bandwidth impact analysis, and throughput capacity factors that you must consider.

There can be definitive trade-offs between storage required, CPU consumed, network bandwidth consumed, and achieved replication throughput. In the planning stages it is a good idea to consider these aspects and understand both the available threshold capacity as well the relative priorities of each aspect.

Connectivity Possibilities

The Capture program is always self-contained to a database, subsystem, or data sharing group, and must be able to connect to the source server database. The Control Center workstation must be able to connect to source and target server databases to perform its tasks, and the Apply program component must communicate with both the source and target server databases, when these are different.

Where communications are used, the connectivity is always through DRDA or the DB2 Universal Database equivalent. The actual communications software that can be used to support the DRDA connectivity varies according to the platforms being connected. Between DB2 Universal Database databases, the choices are TCP/IP, SNA, NetBIOS, and IPX/SPX. DB2 Personal Connect Edition (PCE) is required for connections between DB2 Universal Database databases and DB2 for MVS, DB2 for VSE, or DB2 for VM. TCP/IP or SNA can be used with DB2 for MVS 5.1 and PCE 5.1. All other connections use SNA only.

The more layers of emulation used, LAN bridges added, or router linkups required, the more restricted the replication performance will be. Planning for both current and future needs is essential.

Communications resources can be a major factor in a replication design that involves staging data at a server different from the source database. For example, in a mobile replication scenario between DB2 for MVS 5.1 and DB2 for OS/2, the best connectivity scenario might be to run TCP/IP over a modem link between the remote OS/2 and an AIX staging platform running Apply for AIX. The AIX database would be connected to DB2 for MVS 5.1 through PCE and TCP/IP.

Bandwidth Impact Analysis

IBM Replication is designed to allow for low impact to the network. For example, it allows for the replication of changed data only, supports a data staging arrangement, provides for summarization at the source, can be scheduled to run at off-peak times, and uses DRDA, which enables high-speed, secure data delivery. However, replicating data is not free, and one of the key costs is in bandwidth. So, what is IBM Replication going to do to your network?

The Control Center requires a small amount of capacity. However, the Control Center impact is limited to set up and maintenance of the replication objects.

The Apply program task requires network capacity if the target server and the source server are not the same database or subsystem. In general, the capacity required depends on the volume of data to be applied, the timing window available in which to apply the data, the desired currency of the target data, and the bandwidth installed or to be installed. For example, if a batch program generates many megabytes of change data and the data must be applied to the target system within 30 minutes, the bandwidth requirements will be higher than if the target can be up to 24 hours out of date. The Apply program could then be scheduled to use surplus capacity during periods when network traffic is lighter. For more efficient use of the network capacity, the Apply program is usually installed at the target server so that it can pull the data from the source server. For a more detailed discussion of the differences in pull versus push design, see “Pull Versus Push Apply Design” on page 32 .

Remember that the Apply program is an SQL application and is therefore subject to all of the influences with which any SQL application must handle. Given these factors, the best indicators of likely performance are often found outside the IBM Replication area, in general distributed relational database studies. [4]

Throughput Capacity

Many individual factors influence the throughput possible with IBM Replication. The most important factors include:

- Network bandwidth and tuning
- Source and target machine CPU capacity available
- Database tuning and contention factors
- Change volume and work load mix
- Frequency of the Apply program and number of subscriptions
- The number of Apply program instances used
- Apply program pull versus push approach

Given the complex set of variables involved, you cannot accurately predict the throughput that might be achievable in a given system. At the same time, a feasibility

[4] See the following sources for detailed performance measurements:

- *DDCS/2 to DB2 Performance Benchmarks*
- *Getting Started with DB2 Stored Procedures*
- Web site: http://www.csc.ibm.com/advisor/library/

study would normally need to include some estimation of the potential throughput that is possible.

One way of looking at throughput estimation is to break it down into two parts (assumes a remote pull configuration):

- Remote block fetch of x number of rows — stored into a file (or in MVS, into memory, which of course is the fastest)
- Insert/update/delete of x number of rows in the copy server database

A feasibility study normally includes some estimation of the potential throughput that is possible, and developing a prototype is recommended to verify the throughput in an environment that reflects production conditions.

Security and Authorization Requirements

Security for IBM Replication is a matter of database security. The entire system is table driven, and therefore security of all IBM Replication objects involves database security mechanisms.

Each database has an administrator, who requires sufficient privileges to define replication sources and targets. Additionally, the Apply program uses a qualifier that must be coordinated, but the same user ID can be used to run multiple Apply program instances.

Authorization Requirements for Administration

During the initial defining of replication sources and subscriptions, many tables are created. Depending on the platform, table spaces or dbspaces might also be created. All of these actions require a fairly high level of database privilege, and therefore you should plan on having at least one user ID that acts as the replication administrator and has the authority to create objects and bind plans on each of the replication databases.

The administrator user ID must be a valid logon ID at both the workstation where the Control Center is installed and the source and target sites. The administrator user ID can be used as the user ID running the Capture program or the Apply program, but this is not a requirement.

Authorization Requirements for the Capture Program

The user ID that runs the Capture program must be able to access the system catalog tables, be able to access and update all IBM Replication control tables that are built at the source database, subsystem, or data sharing group, and have execute privileges on the Capture program plan.

The Capture for MVS load library must be APF authorized.

To run the Capture program, you need the following privileges:

- For DB2 for MVS, you need SYSADM, DBADM, or SELECT privileges for SYSIBM.SYSCOLUMNS and SYSIBM.SYSTABLES. You also need the authority to run the Capture program, which includes:
  - EXECUTE privilege on the Capture for MVS plan, ASNLPLAN
  - TRACE privilege
  - MONITOR1 and MONITOR2 privileges
- For Capture for VSE and VM, you must have DBA authority.
- For DB2 Universal Database, you must have DBADM or SYSADM authority that meets all requirements for defining a replication source.

Authorization Requirements for the Apply Program

The Apply program user ID must be a valid logon ID on the source, control, and target servers, and the workstation where the Control Center is installed. The user ID that runs the Apply program must be able to access the replication source tables; access and update all IBM Replication control tables that are built at the source and target database, subsystem, or data sharing group; and update the replication target tables. This user ID must also have execute privileges on the Apply program plan. With the proper authorization, any user ID can run any Apply program instance. The restriction of one user ID per Apply program instance for DPROPR V1 has been removed. On DB2 for MVS, the Apply for MVS load library must be APF authorized.

For DB2 for MVS, you need to have SYSADM, DBADM, or CREATETAB and CREATETS privileges at the source, control, and target server.

For DB2 Universal Database, you must have DBADM or CONTROL or SELECT authority that meets all requirements for defining a replication source.

An Apply program running on DB2 Universal Database might require a password file to connect to the source or target server. For an explanation of configuring security when the Apply program is running on DB2 Universal Database, see the Capture and Apply chapter in this book for your platform.

Customization Requirements

The Control Center creates and populates the replication control tables through SQL files. You can customize these files outside of the Control Center before using the Control Center for replication administration. Most customers need to customize the replication control tables for their site.

The Control Center also creates and populates the replication target tables using SQL generated by replication requests, such as defining a replication source. You can store the SQL in a file and then edit and run it later. Most customers need to customize the control information for replication sources and subscriptions.

See “Working with Customized Replication Control Tables” on page 88 and “Customizing and Running Replication SQL Files” on page 89 to learn how to customize SQL files for your site.

Chapter 6. Application Planning

This chapter helps you evaluate your application level requirements against product capabilities. The term *application level requirements* refers to the requirements of the target application, specifically the type of data required in the target tables (including any subsetting, transformations, or enhancements to the source data), the currency of the data copies, and the consistency of the data copies.

This chapter also helps you further refine your requirements through an evaluation of your source-to-target mapping needs. After reading this chapter, you will be able to plan the time required to implement your replication solution. You will understand which requirements are met by basic techniques, and which require more advanced techniques.

Data Transformation Requirements

Source data most likely has to be subsetted, transformed, or enhanced as part of the replication process, particularly in the support of decision support or data warehousing. This section sorts these requirements into those that are easily fulfilled by using the Control Center and those that require direct manipulation of the control tables.

Basic Data Enhancement

The Control Center supports all of the following basic capabilities:

- **User copy target tables** - These are the most common target tables, and are copies of the replication source without an overhead time-stamp column. These tables require a primary key.
- **Point-in-time target tables** - These are plain copies of source tables with timestamp columns added to specify when updates were made. These tables require a primary key.
- **Base aggregate target tables** - These tables are used if you need to summarize your source table contents on a regular basis. For example, they could be used to track a daily inventory of items from a source table.
- **Change aggregate target tables** - These tables are used if you need to summarize the results of the changes made between each Apply program refresh operation. They could be used on their own, where daily summarizations of change events are required, or as input to an application that must apply the summarizations to a cumulative table.
- **Noncondensed CCD target tables** - These tables are used for history tables or audit trail tables. They can contain changes only, or they can include the initial state of the source table.
- **Condensed CCD target tables** - These target tables can be used as an alternative to a point-in-time copy when it is important to keep the last state of any record, (that is, no records are to be deleted). A delete is processed as an update, with the IBMSNAP_OPERATION column acting as the logical delete flag.

- **Updateable replica target tables** - These target tables can be used as sources for copying data back to the replication source table and are used in update-anywhere scenarios.
- **Column subsetting** - If you want only a subset of the source columns to be copied, use the Control Center. All target table types support vertical fragmentation.
- **Row subsetting** - You can split a single source table into various target tables based on the source column contents; for example, department number or region code; or you specify certain source rows while defining replication subscriptions. All target table types support row subsetting. You might need to define a view replication source, in order to bring together all the columns needed to define the row subset.
- **Before and after images** - Both before and after images can be defined in replication sources and subscriptions. Before images do not make sense with a base aggregate target table type. All other target table types can make use of before-image columns.
- **Column renaming** - Source site column names can be renamed for point-in-time and user-copy target table types.
- **Computed columns** - New columns can be derived from the existing source columns on the basis of SQL expressions. New columns can be aggregate functions such as COUNT or SUM in the case of aggregate target table types, or simple derivations in the case of all other target table types.

  User-defined functions for DB2 Universal Database can be specified by using the computed column facility in the Control Center.
- **Triggers** - On DB2 Universal Database, triggers can be defined on the target table such that row updates on the target can trigger other processing, such as data transformations.
- **Before and after run-time processing** - Run-time statements perform special processing just before or after changes are replicated. For example, data transformation can be applied to source data before replicating the data or to target data after it is copied.

  Depending on the DB2 platform, user processing involving logic might be invoked through the SQL before and after processing:

  – Calls to stored procedures can be made through a before or after processing statement, or triggers can be used to do more complex transformations.

  – On DB2 Universal Database, before or after SQL statements can invoke user-defined functions through the use of triggers.

Advanced Data Enhancement

The transformations listed below require manipulation of the IBM Replication control tables outside of the Control Center. The techniques required to implement these transformations are not discussed in this book. For planning purposes, when you find that you will use many of these techniques in the replication scenario, you should allow additional implementation time.

- **Data consolidation** - Union scenarios can be used to consolidate identically structured data into a common consolidated target table.
- **Outer join of source tables** - Join scenarios can be used when one or more of the sources are subject to change. An outer join can be implemented with multiple subscriptions to a single table.
- **Poorly structured data** - Many older applications often have poorly structured data; for example, conflicting sources for common data items, no primary keys defined, poorly normalized tables, and heavily encoded data values.

  Although IBM Replication might be able to handle these individually, there are some extreme situations where it is not useful to attempt replication before some manual data cleansing is done.

  General rule: - If the data transformation can be expressed by using SQL, IBM Replication will almost certainly be able to execute it. The SQL is introduced either as part of the SELECT statement that does the copying or as part of the user-specified SQL before and after statements. This SQL might include procedural logic by including database triggers, user-defined functions, and stored procedure calls as described in "Basic Data Enhancement" on page 69.

  IBM Replication can also be supplemented with procedural code in user-written programs. For example, a program could be scheduled to manipulate data in the CD or CCD tables before execution of the Apply program. Or, if the changes are so radical that the data cannot be changed in place, you could go as far as writing a program to read the data changes from the CD or CCD tables and then applying them to the target tables yourself. (See "Defining SQL Statements or CALL Procedures for the Replication Subscription" on page 116 to learn about running SQL and Call procedures with replication subscriptions.)

Auditing Requirements

Auditing requirements relate to the need to track histories in terms of before and after comparisons and identification of changes by time and updating user ID.

IBM Replication supports auditing in the following ways:

- **Before and after images** - When you define replication sources, you declare whether or not before or after image columns of the updated rows are available to subscribers.
- **Maintenance of history** - A noncondensed CCD table holds one row per update, insert, or delete operation, thus building a history.
- **Transaction identification** - Each captured change row is automatically stamped with control columns, which are available for audit usage. Two of these columns are the approximate commit time of the changed row at the source server and the operation code (I=Insert, U=Update and D=Delete).

If more user-oriented identification is required, columns for the DB2 correlation ID (if DB2 for MVS is the source database) and the primary authorization ID can be requested in the copy table.

Mobile Replication Requirements

Mobile replication requirements center around having copies of centrally located tables on mobile clients. Special considerations arise with this approach, primarily from the need for the target site to initiate and control the replication operation. IBM Replication supports mobile replication requirements in the following ways:

- **On-demand execution of the Capture and Apply programs** - In an unpredictable mobile environment, the mobile client must initiate and control the connections. To reduce communication costs, the duration of phone connections must be minimized and the mobile client is not continuously in operation. The Capture and Apply programs must operate only for the duration within which all the accumulated transactions are captured and copied to or from the centrally located tables. When the replication sources and subscriptions are defined on the mobile client, the Capture and Apply programs are independent of all other IBM Replication components. They can be run in an on-demand mode whenever required to dynamically connect to the target site.
- **Push and pull options** - If you must upload new or changed data from the mobile clients to the replication source site, the Apply program on the mobile client automatically pushes the data to the replication source site.
- **Changed data retention** - Because of unpredictable connection arrangements, you must ensure that all changed data is retained until each of the mobile clients has replicated its data. The best way to configure IBM Replication for optimized data retention is to use a CCD table, and possibly locate it on a staging server. Two Apply programs are needed. One Apply program moves the data from the CD and UOW tables to a CCD table so that the Capture program can prune the changed data on a regular basis. Another Apply program, on an intermediate server, moves the data from the CCD table to the mobile client. This configuration is not available for update-anywhere replication.

Replication Logical Partitioning Key Considerations

For an application that updates the target primary key column, IBM Replication provides a feature to handle these updates differently. You need to enable logical partitioning key support under any of the following conditions:

- Your source applications update one or more columns of a target table primary key.
- Your source applications update one or more columns of a target table partitioning key (either the target table is a partitioned database managed by DB2 Extended Enterprise Edition (EEE) table or a table in a DB2 for MVS partitioned table space).
- Your source applications update one or more columns referenced in a subscription predicate.

Use the Control Center to specify that the Capture program should support replication logical partitioning key support while defining the replication source. See “Replication Logical Partitioning Key Support” on page 13 to learn more about this option. See “Defining a Custom Replication Source” on page 95 to learn how to enable support for logical partitioning keys.

Data Sharing Considerations

You can implement replication in a data-sharing environment. In a data-sharing environment, you can run one Capture program per source data sharing group and one or more Apply programs per target data sharing group.

Extended Enterprise Edition Considerations

The Apply program and the replication administration function of the Control Center support replication to EEE target servers. Replication support for logical partitioning keys provides the ability to replicate data to partitioned tables with the Capture program option to capture updates as separate DELETE and INSERT operations.

To replicate to a EEE target server, you must select the **Changed data for partitioned key columns captured as delete and insert** option while defining the replication source from the Control Center.

Although the replication logical partitioning key support allows you to replicate to EEE servers, it is not directly related to the DB2 EEE logical partitioning key.

Data Restrictions

Currently, IBM Replication cannot replicate certain types of data. Here is a list of the major restrictions of which you should be aware:

- **Archive log access restrictions with DB2 for MVS V3.1, and DB2 for VSE and VM**

  IBM Replication cannot replicate data that is held only in a DB2 archive log unless you are using DB2 for MVS V4.1 or higher or DB2 Universal Database.

- **Data compression restrictions with DB2 for MVS**

  IBM Replication can replicate data that is compressed through DB2 software or hardware compression on DB2 for MVS V4.1 or higher if the compression dictionary that was used to compress the row is available. Before issuing REORG for compressed replication sources, perform one of the following tasks:

  – Ensure that the Capture program has completed capturing all of the existing changes.

  – Use the KEEPDICTIONARY option on the REORG command to preserve the existing compression dictionary.

  IBM Replication cannot replicate data that is compressed through EDITPROCs or FIELDPROCs.

- **Utility program restrictions**

  IBM Replication cannot capture the data updates made by any of the database utilities. This restriction applies also to data loaded with an option of LOAD RESUME LOG YES.

- **Data encryption restrictions**

  IBM Replication cannot replicate data that is encrypted.

- **Restrictions when using DB2 Extended Enterprise Edition and Data Sharing Environments**

  - There should be no more than one Capture program per sysplex in a data-sharing environment.

  - The Capture program does not support EEE; user tables on a EEE server cannot be defined as replication sources. The Control Center rejects attempts to define replication sources at an EEE server.

- **Data type restrictions**

  Based on the current level of IBM Replication described in this book, here is a full list of the data types that cannot be replicated under any circumstances:

  - Binary large objects (BLOBs)
  - Character large objects (CLOBs)
  - Double-byte character large objects (DBCLOBs)
  - Any column on which an EDITPROC has been defined
  - Any column on which a FIELDPROC has been defined
  - Any column on which a VALIDPROC has been defined
  - Binary data types with precision

  Here is a list of the data types that can be replicated only under certain circumstances:

  - Long variable character string (LONG VARCHAR) requires that the source and target tables are in DB2 for MVS, DB2 for VSE, or DB2 for VM.

  - Long variable graphic (LONG VARGRAPHIC) requires that the source and target tables are in DB2 for MVS, DB2 for VSE, or DB2 for VM.

  - Any user-defined type (distinct data type in DB2 Universal Database) will be converted to the base data type before replication.

Data Currency Requirements

Data currency reflects how up to date you want the copy tables to be. IBM Replication makes it very easy to control the Apply program processing interval and therefore the currency of the data. It is sometimes very tempting to start out wanting your copies to be as current as possible. In a test environment, you might be setting all frequency intervals to the minimum. When moving to production, it is advisable to start with a mid-range frequency value and “tune” your system from that point.

Although customers have devised many creative ways of controlling the replication subscription timing mechanism, the most typical ways are:

- **Interval timing**

  This is the normal mode of operation. In general, replication subscriptions work on a subscription-level interval basis; that is, you specify a refresh interval to be continuous or between 1 minute and 1 year for the replication subscription and all of its members. The interval can be set and changed by using the Control Center or, standard SQL against the subscription set control table.

- **Event-based timing**

  The copy operation can be carried out as a result of an application or user action. This can be done by specifying an event name when you define the subscription in the Control Center, and then populating the subscription event control table with the event name when it is time to trigger the replication subscription. Event-based timing can be used together with interval timing for the same subscription.

You define subscription timing with the Subscription Timing notebook from the Control Center, and control the timing using time-based and event-based scheduling. These timing options can be used together or independently of each other. For example, you can set an interval of one day, but specify that an event can also trigger the subscription cycle.

The simplest method is to use the time-based frequency and allows the subscription to be run at a set interval (relative timing) or continuously. The entire subscription set is replicated on the same frequency. Relative timing allows you to specify an interval in minutes, hours, days, or weeks at which the subscription set is replicated.

Choosing a relative timing interval does not guarantee that the frequency will be exactly at that interval. When you decide on the refresh interval for the replication subscription, you should determine whether it is possible to refresh all tables in the replication subscription within the given interval. To check this, determine the amount of data that the Apply program is likely to select for each interval and estimate the time that it will take to copy the data.

Continuous timing allows you to have the subscription set replicated as frequently as the Apply program can replicate the data.

Setting an event-based frequency allows you to use an event to trigger replication of the replication subscription by the Apply program. You supply an event name while defining the replication subscription with the Control Center and then have an application or a user populate the subscription event control table with the event value. When the Apply program detects the event value, it begins the process of replicating the replication subscription. The Control Center does not manage the triggering of the Apply program. You are responsible for populating the event table.

Data Consistency Requirements

IBM Replication always guarantees table consistency, that is, all data in the target table is actually committed in the base table. IBM Replication also provides consistency across sets of tables with the replication subscription, which can have multiple, related members. These members are each source and target pairs and the members are usually bound by transaction relationships.

The replication subscription helps maintain referential constraints that exist among a set of replication source and replica tables. The replication subscription also limits the boundary for cascade rejection due to RI violation or update collision. Finally, replication subscriptions keep the source-to-target definitions for all the components of a view target table together. The replication subscription should contain all target tables that are related, such as with RI constraints.

Replication Subscription Rules

The following section describes the rules and constraints you need to be aware of in planning and defining a replication subscription.

- If any member of the replication subscription requires full-refresh copying for any reason, the entire set is full refreshed. Full-refresh copying occurs only from the replication source to the replica, not the other way around.
- Each replica target table must be of the same generation as all the other replicas in the replication subscription and come from the same replication source table.
- A single synchpoint is maintained for the replication subscription to indicate the copy progress for the entire replication subscription.
- Collision detection levels must be specified while defining a replication source and affects the processing performed on a replica table in the replication subscription.
- Target tables with RI constraints must be full-refreshed by an outside means using the ASNLOAD exit to bypass RI checking.
- RI violations cannot be detected by application logic in an update-anywhere environment. Declarative RI constraints must be used.
- Referential constraints should not be defined for read-only target tables.
- All referential constraints that exist among the source tables should be included in the replication subscription replicas to prevent RI violations.
- The first occurrence of any RI violation terminates the current replication cycle. The subscription cycle will be retried automatically, after the transaction is rejected and compensated.

Update-Anywhere Replication

Update-anywhere replication is the replication of changed data from either a replication source table or a replica target table defined in a replication subscription. Unlike one-way replication, the target table (the replica) can be updated and the Capture program at the target server captures the changes and the Apply program pushes the changes back to the replication source at the source server. Updates to all of the replicas in a replication subscription are replicated together.

Assessing the suitability of update-anywhere replication for your application is not so much a matter of looking for replication features as it is deciding what sort of anomalies are tolerable to your applications. Is it acceptable that sometimes updates are lost? If so, then you can turn off conflict detection and save processing cycles. Is it acceptable that occasionally a transaction is rejected and automatically compensated? These are the questions you need to ask, and include in your application design process.

The same-row conflict check done by the Apply program is only as good as the information provided. Updates that have not been captured are unknown to the Apply program, escaping the conflict check. Because the Capture program is asynchronous from the user transactions, there is a cost/risk decision that must be made. The cost is the availability of the table for update. The only way for the Capture program to synchronize with your updating applications is to block your updating applications until it has captured all outstanding updates. Taking table level locks on production tables has a negative impact on your production system.

The alternative to synchronizing with the application is to look at the risk of a conflicting update occurring during the Capture program latency window, and at the risk of losing the occasional update. Some customers say that losing some updates is not a problem (For example, one application was a hotel reservations system where rooms are over booked as routine practice), and others design their update cycle so that there is no possibility of conflict.

Conflict Detection

The replication source is considered the primary, or global table. It can receive updates from replicas, but if there is a conflict, the replication source wins and the replicas' conflicting transactions are rejected. An update conflict occurs when an application and one or more replicas try to update the same version of a row in a table at different locations or when constraints are violated. Direct row conflicts are detected by comparing the key values in the CD tables with the source and target tables. If any match is found, the replica unit of work is marked as rejected in the UOW table and the transaction is compensated.

Conflict detection is provided at three levels: no detection, standard detection, and enhanced detection. Each level has a numerical value and is stored in the CONFLICT_LEVEL column of the register control table. You must decide, based on your tolerance for lost or rejected transactions and performance requirements, which type of detection to use.

None No detection. The value is 0.

Standard The Apply program checks for conflicts within the currently captured information, but with the risk of not seeing updates still held in the log. The value is 1.

Enhanced The Apply program takes a shared table lock on the application table during the conflict check and updating period. The value is 2.

Recovering from Conflicts

By using rejection codes provided in the UOW table, you can identify the before and after row values in the change data table for each rejected transaction. When the Apply program compensates the rejected transactions, you must decide how to handle the rejected transactions. The ASNDONE user exit notifies you when the subscription cycle is completed, regardless of whether there are conflicts or rejected transactions. You can have your application handle the rejected transactions and have the Apply program run your application logic in the ASNDONE user exit. See the Capture and Apply chapter in this book for your platform for more information on the Apply program ASNDONE exit. You can choose to handle the rejections in batch; the change data rows and UOW control table rows for rejected transactions are exempt from normal pruning. They are subject only to RETENTION_LIMIT pruning.

The following suggestions for handling the rejected transactions have not been tested in user environments and should be used as suggestions:

- In a mobile environment, develop part of the application to run at the end of the Apply program's subscription cycle, immediately notifying the user of any conflicts that might be replicated to other servers.
- In a server-to-server configuration, write a batch application that periodically scans the UOW control tables at the replica sites for evidence of rejected transactions.
- Automate the re-running of any rejected transaction, putting them into a proper transaction sequence.

CCD Tables

Because update-anywhere replication requires referential integrity, CCD tables are not used in update-anywhere replication. CCD tables stage data and transactions are condensed. In update-anywhere replication, every transaction update should be replicated to the target server in the original order it occurred, which does not occur with CCD staging tables. Additionally, there may be triggers defined which rely on reading each update. Because you want to synchronize the source and target servers as up-to-the-minute as possible to detect update conflicts, you want to prevent the delays imposed by CCD staging. If you defined an internal CCD at the source server, it is ignored by the Apply program when processing a subscription with a replica as a target.

Recommended Usage

To reduce the risks of conflicts and cost of rejected conflicting transactions, use update-anywhere replication under the following conditions:

Fragmentation by key
Design your application so that the replication source is updated by replicas for key ranges at specific sites.

Fragmentation by time
Design your application so that the table can be updated only during specific time periods at specific sites. The time periods must be sufficiently separated to allow for the replication of any pending changes to be made to the site that is now becoming the master version.

Restrictions

The following list describes restrictions for update-anywhere replication:

- Enhanced conflict detection requires simultaneous availability of the Capture and Apply programs.
- RI violations cannot be detected by application logic in an update-anywhere environment. Use declarative RI constraints.

  If referential constraints that exist among the source tables are omitted from the corresponding set of replicas, an update made to a replica could result in an RI violation when replicating the update to a source table.

Minimizing Contention between Capture and Apply for MVS 3.1 Sources and Targets

Contention can occur when the Capture and Apply programs try to access the same table on DB2 platforms that do not support ISOLATION (UR).

To minimize contention:

- Use the NOPRUNE Capture program option to defer pruning so that pruning activity does not block the Apply program.
- Minimize the number of Apply program instances copying from the same source server. The more Apply program instances that copy from the same source server, the higher the probability that the Capture and Apply programs encounter contention difficulties.
- Minimize the number of Apply program instances copying from the same source table. The more Apply program instances that copy from the same source table, the higher the probability that the Apply program instances encounter contention over access to the pruning control tables.
- Schedule the Capture and Apply programs to run at different times of the day.
- Shorten the Capture program commit intervals (in the tuning parameters table), and lengthen the frequency of the Apply interval (on the Frequency page of the Subscription Definition notebook).

How Locking Affects Contention on Platforms without ISOLATION (UR) Support

Change data tables and UOW tables can be held for a long time; they can be locked in two ways: global locks or a single LOCK TABLE. The first is a global lock, which is a lock against the global critical section table. A single LOCK TABLE, obtained before the table is accessed, is issued.

The Capture program gets the critical section lock in *exclusive* mode before it starts capturing changes. During this time, the Apply program cannot access those tables; the Apply program must get the critical section lock in *shared* mode. After this sequence of change captures (that is, after a scheduled COMMIT), the Capture program becomes inactive, allowing one or more Apply program processes to get the lock in shared mode.

You can decrease the COMMIT_INTERVAL in the tuning parameters table; this forces the Capture program to give up the lock more often, giving the Apply program the opportunity to access the change data table. If you decrease the COMMIT_INTERVAL value to less than 20 seconds while using automatic pruning, you also increase the frequency of pruning while slowing performance.

The benefit of trying to make change data available to the Apply program immediately is outweighed by the increased chance of contention between the Capture and Apply programs.

Part 3. Administering Your Replication System

Administering a replication system involves identifying source tables and views, and defining the location, structure, and subscription cycle of the target tables. This part describes administration concepts and tasks that can be performed using the DB2 Universal Database Control Center.

Chapter 7. Administration Overview

You use the Control Center to define sources and targets for replication, as well as to set the schedule for updating the targets, to specify the enhancements to the target data, and to define any triggers that kick off replication. The administration tasks described in this chapter set up the control information that both the Capture and Apply programs use to capture updated data and then replicate it to the target tables, in the proper format, and at the appropriate interval.

Replication administration has two major tasks: defining replication sources and defining replication targets (subscriptions). Replication sources are DB2 tables, views, or nicknames used as source for copying data to a target table. Replication subscriptions are the specifications for one or more target tables and their location, structure, and timing schedule, as well as any SQL enhancements that are necessary. Other replication tasks include maintaining the replication sources and subscriptions, and cloning them to other servers.

Overview of Replication Administration Steps

When setting up the replication environment, you must use the Control Center to manage the source and target table definitions and the control tables. See “Replication Products” on page 57 for information about how the Control Center is packaged. The following list describes the high-level steps you use to administer your replication objects. See Chapter 3, “Getting Started” on page 39for a sample scenario for most of these tasks.

1. Check, and optionally, update the default settings in the Tools Settings notebook.
2. Review the DPCNTL file for your platform to determine whether you need to customize the control tables for your site.
3. Optionally customize the DPCNTL file for your platform and site requirements. See “Working with Customized Replication Control Tables” on page 88 for more information.
4. Define and manage replication sources. See Chapter 8, “Working with Replication Sources” on page 93 for more information.
5. Define and manage replication subscriptions. See Chapter 9, “Working with Replication Targets” on page 103 for more information.

Once you created the control tables and defined replication sources and targets, you need to configure and run the Capture and Apply programs to begin replicating data.

This chapter describes the following tasks:

- How to navigate through the Control Center to your replication objects
- How to set replication preferences (default settings) in the Tools Settings notebook
- How to customize the DPCNTL file and create the control tables

- How to customize and run SQL files generated by replication requests
- How to specify ordinary and delimited identifiers

Navigating to Your Replication Objects with the Control Center

You can access your replication sources and targets through the Control Center interface. There are three containers in the Control Center for organizing the objects that you use to set up and maintain your replication environment:

Tables folder
The folder containing user-defined DB2 relational tables.

Replication Sources folder
The folder containing tables that have been defined as replication sources: DB2 tables, internal tables, views, or target tables redefined as sources for replication.

Replication Subscription folder
The folder containing subscription definitions for copying updated source data to target tables.

Figure 13 shows the Control Center. The folders you use for replication tasks are in the tree structure on the left side of the window, called the object tree.

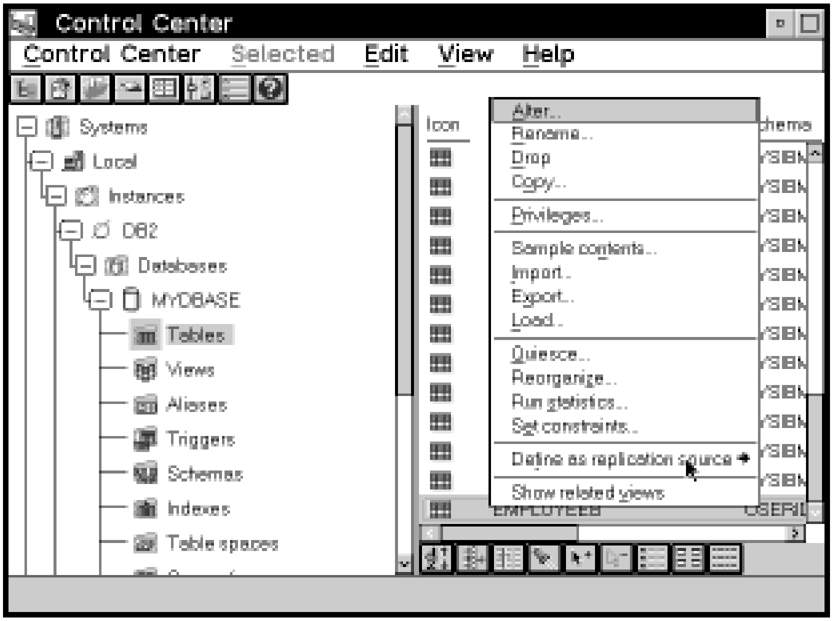


Figure 13. The Replication Folder Objects in the Control Center Window. The Replication folder objects are used to administer your replication environment.

Each object also has a folder with the actions that can be performed with the object. In Figure 13, a table object has been selected and the pop-up menu is displayed with the **Define as replication source** -> **Custom** action selected.

To access the replication folders and objects:

1. From the Control Center window, expand the object tree until you find the replication folder that you need to use: **Tables**, **Replication Sources**, or **Replication Subscriptions**.
2. Click on the folder. All objects contained in the folder are displayed in the pane on the right side of the window (the contents pane).
3. Click mouse button 2 on the object you want in the contents pane and select an action from the pop-up menu. A window related to the action opens or a process is launched.

In addition, some actions can be performed against an entire folder. For example, you can refresh the **Replication Sources** folder.

Replication Restrictions for Extended Enterprise Edition

Replication features are not available in DB2 Extended Enterprise Edition (EEE) V5. EEE table objects cannot be defined as replication sources or replication subscriptions. This means that replication menu choices are not available on EEE table, replication source, or replication subscription objects from within the Control Center.

Configuring the Control Center for Non-DB2 Universal Database RDBMs

If you are connecting to a DB2 for MVS, DB2 for VSE, or DB2 for VMsource or target database server from the Control Center, you must bind the database to the Control Center.

To bind the database:

1. Change to the directory where the Capture program bind files are located, which is usually *drive*:\SQLLIB\BND.
2. Create and bind the DB2 for MVS, VSE, or VM package to the DB2 database by entering the following command:

   ```
   DB2 BIND @DDCSxxx.LST ISOLATION UR BLOCKING ALL SQLERROR CONTINUE
   ```

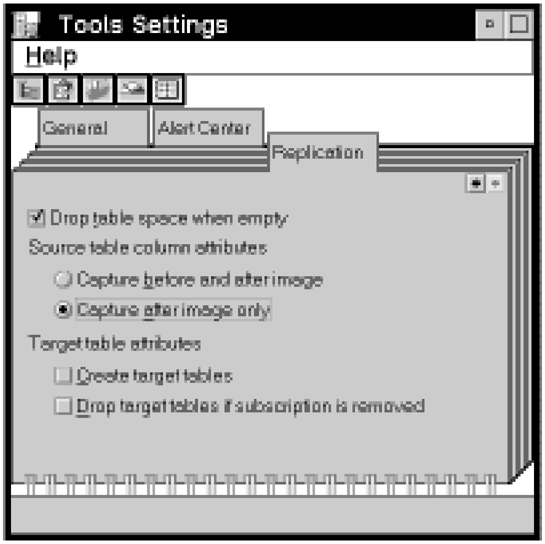
   Where *UR* specifies the list in uncommitted read format for greater performance and *xxx* specifies the platform name: MVS, VSE, or VM.

   If your system does not support UR, substitute CS (cursor stability format) instead.

Additionally, you might need to perform an explicit connect from the Control Center. If the password is different than the local logon ID and password at the Control Center workstation, you must explicitly connect to the database server using the **Connect** menu choice from the pop-up menu for the icon of your DB2 for MVS database.

Setting Replication Preferences in the DB2 Tools Settings Notebook

The Tools Settings notebook contains default preferences for the DB2 Universal Database administration tools. You can set replication default values on the Replication page of the notebook, as shown in Figure 14 on page 86. These default values are used for all replication activities administered by the Control Center.



☑ Drop table space when empty

Source table column attributes

○ Capture before and after image

● Capture after image only

Target table attributes

□ Create target tables

□ Drop target tables if subscription is removed

Figure 14. The Replication Page of the Tools Setting Notebook. The Replication page of the Tools Settings notebook contains default preferences for replication.

The default values for replication sources and subscriptions are:

Table spaces
Change data table and target table spaces are dropped when empty.

Replication sources
Before- and after-image columns are captured.

Replication targets

- Target tables must be created by the user.
- Target tables are not dropped if the subscription is removed.

To set replication preferences in the Tools Settings notebook:

1. From the Control Center, select the **Tools Settings** icon in the tool bar as shown in Figure 15 on page 87.

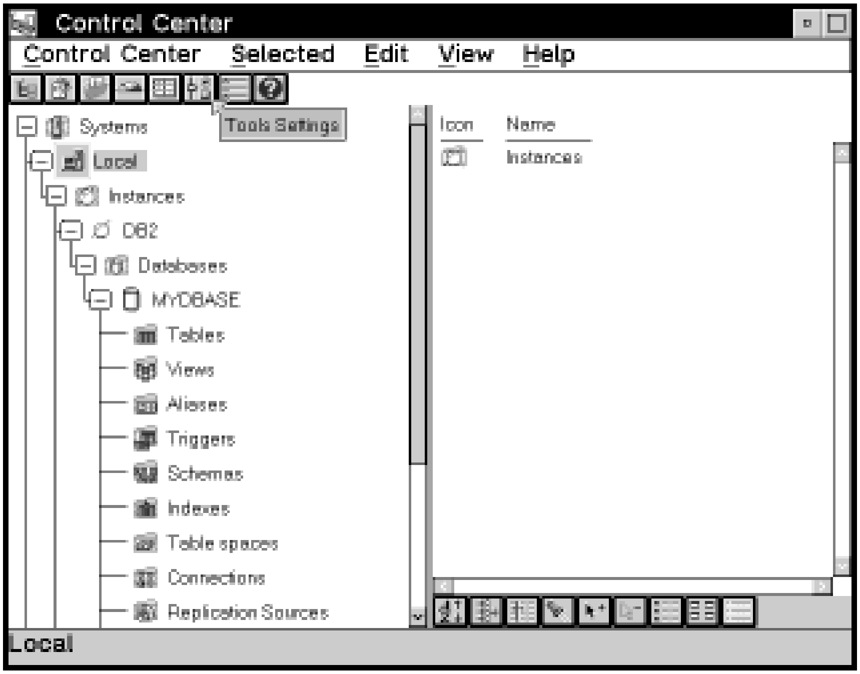


Figure 15. The Tools Settings icon. The Tools Setting icon opens the Tools Setting window where you can set replication preferences.

The Tools Settings notebook opens.

2. Click on the **Replication** tab to open the Replication page.
3. If you want both the table space for the change data table (at the source server) and the table space for the target table (at the target server) to be dropped when the table space is empty, select the **Table spaces are dropped when empty** check box.

   When selected, table spaces are dropped for the specified table types on MVS, OS/2, Windows NT or 95, AIX, HP-UX, Solaris, VSE, and VM.

4. For capturing before- and after-image columns on the replication source, select a radio button:

   Capture before and after image
   The before image and after image of the source table column is captured and can be replicated to a target table.

   Capture after image only
   Only the after image of the source table column is captured and can be replicated to a target table.

5. If you want the Control Center to create the target table automatically while defining a replication subscription, select the **Create target tables** check box. If the target table is not selected, you need to create your own target table. See “Defining

a Replication Subscription with a User-Defined Target Table” on page 119 for more information about defining and maintaining target tables.

6. If you want to drop affected target table and related control tables when the replication subscription set is removed, select the **Drop target tables if subscription is removed** check box.
7. Close the Tool Settings notebook to save your changes.

Working with Customized Replication Control Tables

The replication control tables are created in two ways:

- By customizing the DPCNTL file for your platform and then running the file before running any actions from the Control Center.
- By running the first replication request at a server from the Control Center to create the default version of the control tables.

If you use the second option, you cannot customize the replication control tables for your site without dropping the existing control tables and customizing the tables as discussed below. The default custom tables are designed for the DB2 Universal Database platforms. If you are running on MVS/ESA, VSE/ESA, or VM/ESA, you *must* customize the replication control tables.

You need to customize the file for your platform before performing any replication action at the server. See comments within the file for tailoring the SQL to run on a specific database platform. You will need to customize the DPCNTL file for the following definitions:

- To define and size DB2 for MVS table spaces and databases for the control tables. The Control Center automatically creates the control tables in the default table space and database, unless you specify a different table space or database.
- To define DB2 for MVS V3.1 padding to support row-level locking.
- To define and size DB2 for VSE or VM dbspaces for the control tables. The Control Center will automatically create the control tables in the default dbspace, unless you specify a different dbspace.
- To tailor the control tables to run on specific platforms because not all definitions are supported on all platforms.
- To define additional indexes for better performance on a specific platform.
- To place control tables in specific DB2 Universal Database table spaces or in a DB2 EEE single node group.

When you create customized control tables, you need to customize SQL files for the CREATE TABLE statements. The DPCNTL file contains the SQL for creating control tables. There is a DPCNTL file for each platform located in the SQLLIB\SAMPLES\REPL\ directory. The files names are:

DPCNTL.UDB
Creates control tables for the DB2 Universal Database platform.

DPCNTL.MVS
Creates control tables for the DB2 for MVS platform.

DPCNTL.VM
Creates control tables for the DB2 for VSE and VM platforms.

To customize the SQL for creating control tables:

1. Open the SQLLIB\SAMPLES\REPL\DPCNTL.*platform_name* file as ASCII in an editor, where *platform_name* is the platform name in the previous list.
2. Read the commented areas for each platform and table.
3. Edit the file for your site or application.
4. Close the file.
5. Run the file by entering the following command from a command line window:

   ```
   db2 -tf dpcntl.platform_name
   ```

When you drop customized control tables, you need to customize SQL files for the DROP TABLE statements. The DPNCNTL file contains the SQL for dropping control tables. There is a DPNCNTL file for each platform located in the SQLLIB\SAMPLES\REPL\ directory. The files names are:

DPNCNTL.UDB
Drops control tables for the DB2 Universal Database platform.

DPNCNTL.MVS
Drops control tables for the DB2 for MVS platform.

DPNCNTL.VM
Drops control tables for the DB2 for VSE and VM platforms.

To customize SQL for dropping the control tables:

1. Open the SQLLIB\SAMPLES\REPL\DPNCNTL.*platform_name* file as ASCII in an editor, where *platform_name* is the platform name in the previous list.
2. Read the commented areas for each platform and table.
3. Edit the file for your site or application.
4. Close the file.
5. Run the file by entering the following command from the command line window:

   ```
   db2 -tf dpncntl.platform_name
   ```

Customizing and Running Replication SQL Files

You have the option to run a replication task immediately or save the SQL generated by a replication action in an SQL file to be updated and run at a later time. The SQL files can be customized for large scale replication actions such as defining subscriptions, or customized for the application beyond implementations supported by the Control Center.

SQL files allow you to customize the replication tasks for your site or application and give you flexibility as to when and how you run the files. You can:

- Batch the definitions together.
- Defer execution of the replication action until a specified time.
- Create libraries of SQL files for backup, site-specific customization, or to run stand-alone at distributed sites, such as in a mobile environment.

If you save the definitions of a large replication subscription in an SQL file, you can replay the definitions as necessary.

Types of customizing that you might do include:

- Creating multiple copies of the same replication action, customized for multiple servers, and run a large scale action.
- Customizing CD table names.
- Defining the location for CD tables (DB2 for MVS database, DB2 Universal Database table spaces, DB2 for VSE and VM dbspaces).
- Creating and sizing the table spaces, databases, or dbspaces of the CD tables.
- Defining site-specific standards.

After completing a replication task, such as defining a replication source, the Run Now or Save SQL window opens, giving you a choice to run the SQL for the task now, or save it so you can edit it and run it later.

To defer the replication action SQL files:

1. Select the **Save SQL to file and run later** check box in the Run Now or Save SQL window.
2. Click on **OK**. The Save SQL File window opens.
3. Type the name and file path for the file.
4. Click on **OK** to close the window and save the SQL file. A confirmation window indicates that the file has been saved.

After you defer the SQL file, you can customize the file for your application.

To customize a replication action SQL file:

1. Open the SQL file that you saved as ASCII in an editor.
2. Edit the file for your site or application.
3. Click on **OK** to close and save the file.

You can run deferred files from the Control Center.

To run deferred replication action SQL files:

1. In the Control Center, select the replication folder for the object type that you are running the SQL file for. For example, if you saved and customized an SQL file to define a replication source, select the **Replication Sources** folder.
2. Click mouse button 2 to display the pop-up menu for the folder and select the **Run SQL files** menu choice. The Run SQL File window opens (Figure 16).

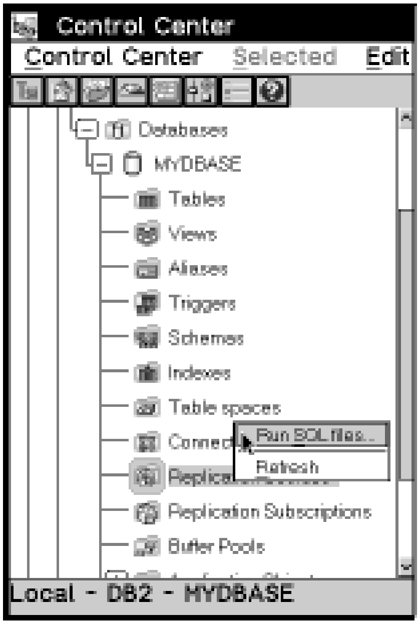


Figure 16. The Run SQL files menu choice. The Run SQL files menu choice is used to run deferred SQL files for the replication object you select.

3. Select the directory path and file name of the SQL file.
4. Click on **OK** to run the file. A confirmation window opens indicating when the SQL file has run successfully.

You can also run the SQL file from the DB2 Universal Database Command Line Processor (CLP) or the DB2 Universal Database Command Center.

Ordinary and Delimited Identifiers

Many of the fields in the Control Center allow two types of SQL identifiers: ordinary identifiers and delimited identifiers. Many replication values, such as table names, require that the case of the value match the name of the object. The online help and the tasks in this book specify which fields accept both ordinary or delimited identifiers.

An ordinary identifier:

- Must start with a letter
- Can include uppercase letters, numbers, and underscores (_)

- Cannot be a reserved word

Note: If you type a lowercase letter as part of an ordinary identifier in a field, the lowercase letter is stored as an uppercase letter.

A delimited identifier:

- Must be enclosed within quotation marks (")
- Can include uppercase and lowercase letters, numbers, underscores (_), and spaces
- Can contain a quotation mark, represented by two consecutive quotation marks

Note: Leading spaces are stored as part of a delimited identifier, but trailing spaces are ignored.

To name a target table `WKLYSAL`, you can type: `WKLYSAL` or `wklysal` or `Wklysal` in the **Target table** field. The lowercase letters are changed to uppercase and the table name is stored as `WKLYSAL`. If you want the name of the table to be `Wkly Sal`, type: `"Wkly Sal"` in the **Target table** field. The quotation marks are recognized as the delimiters of the identifier, and the table name is stored as `Wkly Sal`.

If you want the name of a column to be `"Nickname"`, type: `"""Nickname"""` in the **Column name** field. The outermost quotation marks are recognized as the delimiters of the identifier and each pair of consecutive quotation marks represents one quotation mark. The column name is stored as `"Nickname"`.

Chapter 8. Working with Replication Sources

This section describes replication sources and how to work with them in your replication environment.

What Is a Replication Source?

A replication source is a table used as a source for copying data to a target table. Replication sources can be the following types of tables:

User table
A table created and used (read and updated by an application) outside of replication. Updates to the table can be captured and copied to a replication target table.

Replication target table
Read-only target tables that have been redefined as sources for replication.

Consistent change data (CCD) tables
A staging table, a target table type that has been redefined as a source table for staged replication.

External CCD table
A consistent change data table that has been populated from an external source such as an IMS segment.

Replica table
An updateable target table that is used to replicate data back to the origin table.

When you define a table as a replication source, a replication source object is created and a row is inserted into the register control table, making it available as a source table for replication subscription. Unless you specify full-refresh copying, the source table is enabled for change data capture: the table is checked for the Data Capture Changes option of the CREATE TABLE SQL statement. If it is not already enabled, the source table is altered to enable the option.

Defining Replication Sources

To define a replication source, use the Control Center Object Tree to navigate to the **Tables** folder. The **Define as replication source** pop-up menu choice allows you to define a table object as a replication source. Figure 17 on page 94 shows the pop-up menu for the table object and the **Define as replication source** menu choice.

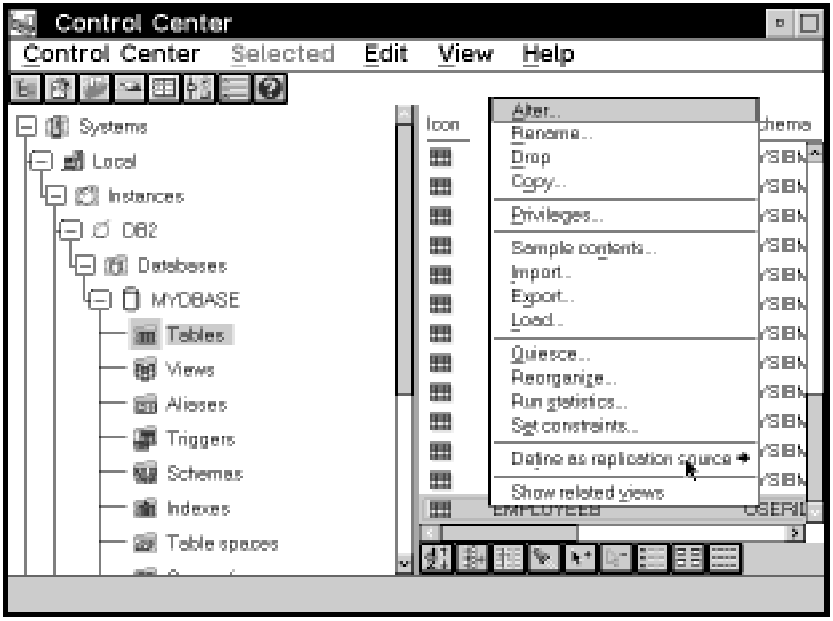


Figure 17. The Table object pop-up menu. The Table object pop-up menu contains the Define as Replication Source menu action.

You can define replication sources using the **Quick** or **Custom** pop-up menu choices. **Quick** allows you to define a replication source using the defaults set in place. **Custom** allows you to customize the defaults, such as specifying any columns that should not be captured for update replication.

When you finish defining the source, you can select whether to run the SQL that defines the source, or defer and run the file later. When you run the SQL file to define the source, an object is created in the **Replication Sources** folder. The table can now be defined in a subscription set.

For data restrictions when defining replication sources and targets, see “Data Restrictions” on page 73.

Defining a Replication Source with the Default Values (Quick)

Use the **Define as Replication Source -> Quick** menu choice, available from a table object on the right side (contents pane) of the Control Center, to define tables as sources for replication using the default values.

To define a default replication source with the Quick menu choice:

1. Check the default settings in the Tools Settings notebook on the Replication page. See “Setting Replication Preferences in the DB2 Tools Settings Notebook” on page 85 for a list of the setting defaults and instructions for opening the notebook and change the settings.

Additional default values can be changed only by using the **Custom** menu choice and selecting choices on the Define as Replication Source window.

- All source columns are available for data capture.
- Partition key columns are captured as UPDATE statements by the Capture program.
- If the source table will be used in update-anywhere replication, conflict detection is none.

If these default values are not acceptable for your application, go to "Defining a Custom Replication Source."

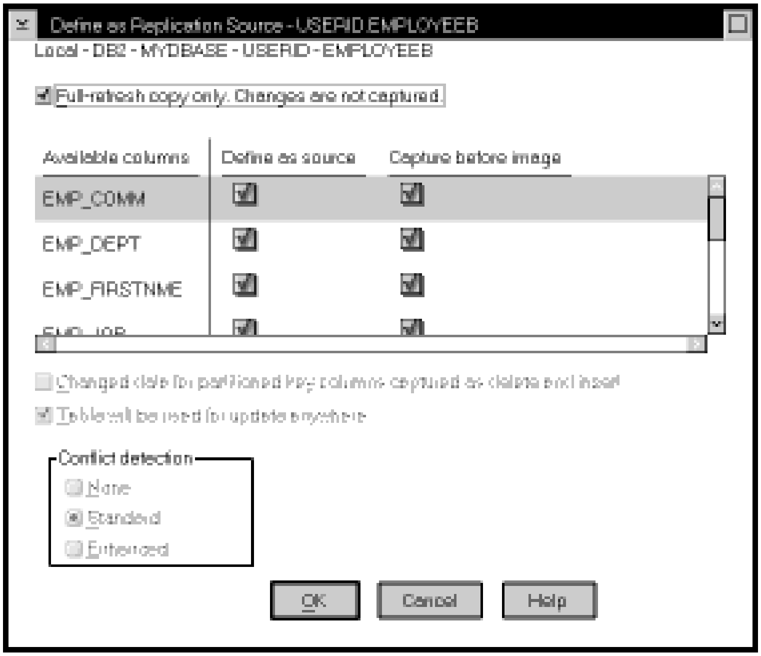
2. In the Control Center, navigate to the **Tables** folder in the source database and select it. The tables in the source database are displayed on the right side of the window contents pane.
3. From the contents pane, click mouse button 2 on the source table icon and select the **Define as Replication Source -> Quick** menu choice to launch the definition of the source. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See "Customizing and Running Replication SQL Files" on page 89 to learn how to submit the request now or later.
4. After you submit the request, verify that the replication source was created by selecting the **Replication Sources** folder in the object tree and selecting the **Refresh** menu choice.
5. To begin capturing changes to the source table, start or reinitialize the Capture program from the command processor window. See the Capture and Apply chapter for your platform in this book for more information about the Capture program.

Defining a Custom Replication Source

Use the Define as Replication Source window to identify tables that are available as sources for replication.

To define a custom replication source with the Custom menu choice:

1. Check the default settings in the Tools Settings notebook on the Replication page. See "Setting Replication Preferences in the DB2 Tools Settings Notebook" on page 85 for a list of the settings defaults and to learn how to open the notebook and change the settings.
2. In the Control Center, navigate to the **Tables** folder in the source database and select it. The tables in the source database are displayed on the right side of the window contents pane.
3. From the contents pane, click mouse button 2 on the source table icon and select **Define as Replication Source -> Custom** to open the Define as Replication Source window, as shown in Figure 18 on page 96.



Define as Replication Source - USERID.EMPLOYEEB

Local - DB2 - MYDBASE - USERID - EMPLOYEEB

☑ Full-refresh copy only. Changes are not captured.

Available columns	Define as source	Capture before image
EMP_COMM	☑	☑
EMP_DEPT	☑	☑
EMP_FIRSTNME	☑	☑
EMP_JOB	☑	☑

□ Changed data for partitioned key columns captured as delete and insert

☑ Table will be used for update anywhere

Conflict detection

○ None

● Standard

○ Enhanced

Figure 18. The Define as Replication Source window. The Define as Replication Source window is used to define replication sources.

4. If you want to specify that the Capture program will not capture changes to the source table and that the source data will be available for refresh copying, select the **Full-refresh copy only** check box.

5. If you want to make a column unavailable for replication, clear the **Define as source** check box for that column. When cleared, the column name becomes unavailable as a source column while defining a replication subscription and cannot be replicated.

   This check box is automatically selected if the **Table will be used for update anywhere** check box is selected.

6. If you do not want to capture the before image of a column, clear the **Capture before image** check box for that column. The after image is always replicated.

   This check box is selected automatically when the **Table will be used for update anywhere** check box is selected.

7. If you want support for logical partition keys, select the **Changed data for partitioned key columns captured as delete and insert** check box. The Capture program treats updates to primary key columns as INSERT and DELETE statements, rather than UPDATE statements.

8. If the source table might be used for update-anywhere replication:

   a. Select the **Table will be used for update anywhere** check box.

When you select this check box, the **Conflict detection** choices are enabled. If you do not select the **Table will be used for update anywhere** check box, the conflict detection level default is None.

b. Select a level of conflict detection.

None Select **None** if you do not want any conflict detection.

Attention: Conflicting updates between the origin table and the replica will *not* be detected.

Standard Select **Standard** if you want moderate conflict detection in which the Apply program searches rows already captured in the replica's change data tables for conflicts. Standard detection is the default value.

Enhanced Select **Enhanced** for conflict detection that guarantees data integrity among all replicas and the origin table. The Apply program locks all replicas in the subscription set against further transactions, and begins detection after all changes prior to the locking have been captured.

See “Conflict Detection” on page 77 for more information about conflict detection.

9. Select **OK** to save the values and close the window. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.
10. After you submit the request, verify that the replication source was created by selecting the **Replication Sources** folder in the object tree and selecting the **Refresh** menu choice.
11. To begin capturing changes to the source table, start or reinitialize the Capture program from the command processor window. See the Capture and Apply chapter for your platform in this book for more information about the Capture program.

Defining an External CCD Table as a Replication Sources

Changes captured within applications or other system tools, such as DataPropagator NonRelational, can also be defined as sources for replication subscription. The external data source must provide a complete CCD table, which you can define as a replication source with the Control Center and use as a surrogate source table. You can create the CCD table outside of the replication tools, have your application populate it, and then define the table as a replication source. See “External Data Sources” on page 29 for more information about external data sources.

To define a CCD table as an external replication source:

1. Create and populate the CCD table with the user data columns in CCD table format, to hold changes that originated from the external data source. See

Chapter 20, “Table Structures” on page 265 for more information about CCD table format.

2. Define the CCD table as a replication source. See “Defining Replication Sources” on page 93.

Defining Join Replication Sources

You can define replication sources that are views of other tables. After defining each replication source table included in the view, you can create a view replication source. The view replication source is then available to be copied to a target table.

You cannot define an existing view as a replication source.

To define a join:

1. Define as replication sources the source tables to be used in the join. See “Defining Replication Sources” on page 93.
2. From the Control Center object tree, click on the **Replication Sources** folder and select the replication sources to be used in the join from the contents pane.
3. Click mouse button 1 and select **Define join** from the pop-up menu. The Define Join window opens, as shown in Figure 19 on page 99.

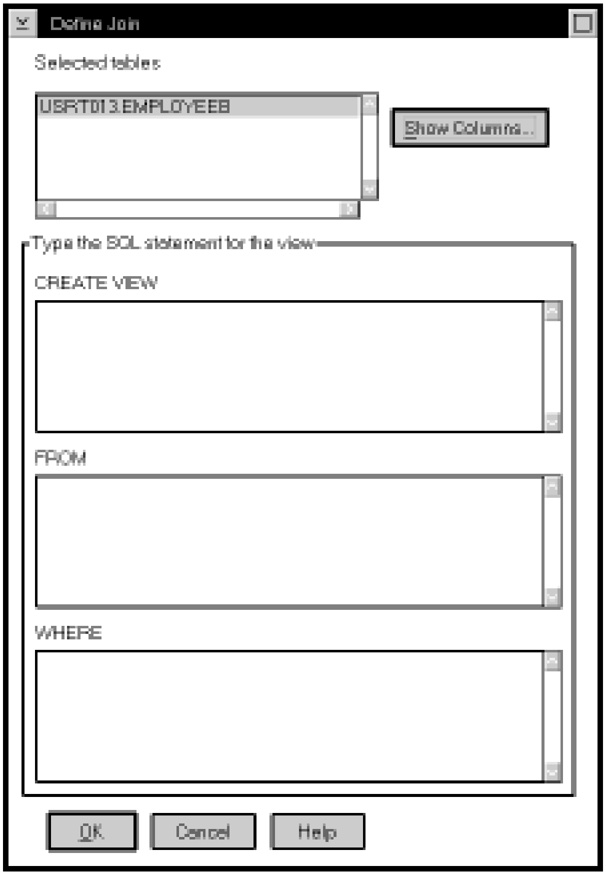


Figure 19. The Define Join Window. You can define joins as replication sources.

4. To view a list of columns available for the join:
   a. Select a table name from the **Selected tables** box.
   b. Select **Show Columns**. The Show Columns window opens and shows all of the columns in the selected table.
   c. Select **OK** to close the window and continue defining the join.
5. In the **CREATE VIEW** field, type the SQL statement for the view. For example:

   ```
   USERID.VIEW_NAME AS SELECT A.COL1, A.COL2, B.COL6, B.COL5
   ```

   This field can contain ordinary or delimited identifiers. See “Ordinary and Delimited Identifiers” on page 91 for more information.

   Do not type the words CREATE VIEW. This part of the statement is automatically supplied during processing.
6. In the **FROM** field, type the SQL statement for the view. For example:

   ```
   TABLEA A, TABLEB B
   ```

This field can contain ordinary or delimited identifiers. See “Ordinary and Delimited Identifiers” on page 91 for more information.

Do not type the word FROM. This part of the statement is automatically supplied during processing.

7. If you want to use a row predicate, type the WHERE clause SQL statement in the **WHERE** field. For example:

   ```
   A.COL1=B.COL1
   ```

   This field can contain ordinary or delimited identifiers. See “Ordinary and Delimited Identifiers” on page 91 for more information.

   Do not type word WHERE. This part of the statement is automatically supplied during processing.

8. Select **OK** to save the values and close the window. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.
9. After you submit the request, verify that the replication source was created by selecting the **Replication Sources** folder in the object tree and selecting the **Refresh** menu choice.

Viewing or Changing Existing Replication Sources

You can view an existing replication source or, if you selected the **Table will be used for update anywhere** check box, you can change the conflict detection level defined for the replication source. All other fields and controls are unavailable for changes after you successfully define the replication source.

To view or change the replication source:

1. If you plan to change the replication source definition, stop or suspend the Capture program. See the Capture and Apply chapter for your platform in this book to learn more about the Capture program.
2. Select the **Replication Sources** folder. The replication sources appear in the contents pane.
3. Select the replication source object you want to view or change from the contents pane and select the **Change** menu choice from the pop-up menu. The Change Replication Source window opens.
4. If the table was made available for update-anywhere replication and you want to change the **Conflict detection** choices, select a radio button for the change.
5. Select **Cancel** if you have not changed the replication source definition. Select **OK** to save the values and close the window. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and

Running Replication SQL Files" on page 89 to learn how to submit the request now or later.

6. To begin capturing changes for the changed replication source, start or reinitialize the Capture program. See the Capture and Apply chapter for your platform in this book to learn more about the Capture program.

Notes for changing the conflict level detection:

If you want to be able to change the conflict detection level, you must have done the following while defining the replication source:

- Selected the **Define as source** check boxes selected for all columns
- Selected the **Capture before image** check boxes for all columns
- Optionally selected the **Table will be used for update anywhere replication** check boxes

Removing Replication Sources

When you no longer need a replication source, you can remove the object from the Control Center and its control information from the control tables.

Attention:

1. Do not delete a replication source while the Capture program is running.
2. Before you delete a replication source, determine which subscriptions use the replication source as a source for copying. Delete the subscriptions before deleting the replication source.

The Control Center drops the replication source table space if it is empty. You can change this default behavior on the Replication page of the Tools Settings notebook so that the table space is not dropped.

To remove a replication source:

1. Stop the Capture program. See the Capture and Apply chapter for your platform in this book to learn how to stop the Capture program.

   Attention: Do not use the SUSPEND command.

2. Select the **Replication Sources** folder. The replication sources appear in the contents pane.
3. Select one or more replication source objects that you want to delete and select **Remove** from the pop-up menu.

   A confirmation window prompts you to confirm that you want to remove the replication sources.

4. Select **Yes** in the confirmation window to delete the replication sources. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later.

See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.

Attention: Before running the deferred remove request, first stop the Capture program.

5. After you submit the request, verify that the replication sources were deleted by selecting the **Replication Sources** folder in the object tree and selecting the **Refresh** menu choice.
6. Restart the Capture program. See the Capture and Apply chapter for you platform in this book to learn how to restart the Capture program.

Chapter 9. Working with Replication Targets

The following section describes replication subscriptions and how to work with them in your replication environment.

What Are Replication Subscriptions?

A replication subscription is a set of definitions that define the relationship between one or more source and target tables, the structure of a target table, and the way the Apply program replicates data to the target table: when, how often, and which enhancements. Each definition is called a member and also defines the type of target table to be created, such as a user copy table or a replica. Data is replicated from all replication source tables to all target tables in the same processing cycle, and all data is replicated from all replica target tables to the replication source table in the same processing cycle.

The replication subscription logically groups one or more definitions that are somehow related; for example, source tables that have RI constraints. For the related tables, the changed data needs to be copied to the target tables at the same time, or in the same subscription interval. See "Data Consistency Requirements" on page 76 for more information about replication subscriptions and RI constraints.

Defining Replication Subscriptions

Use the Define Subscription window to set up sources and targets to define a basic replication subscription. See "Navigating to Your Replication Objects with the Control Center" on page 84 to learn how to navigate to the **Replication Sources** folder. Many of the fields in the Define Subscription window accept both ordinary and delimited identifiers. These fields are identified in the step. See "Ordinary and Delimited Identifiers" on page 91 for more information.

To define a replication subscription:

1. Select the **Replication Sources** folder from the Control Center as shown in Figure 20 on page 104. The replication sources appear in the contents pane.

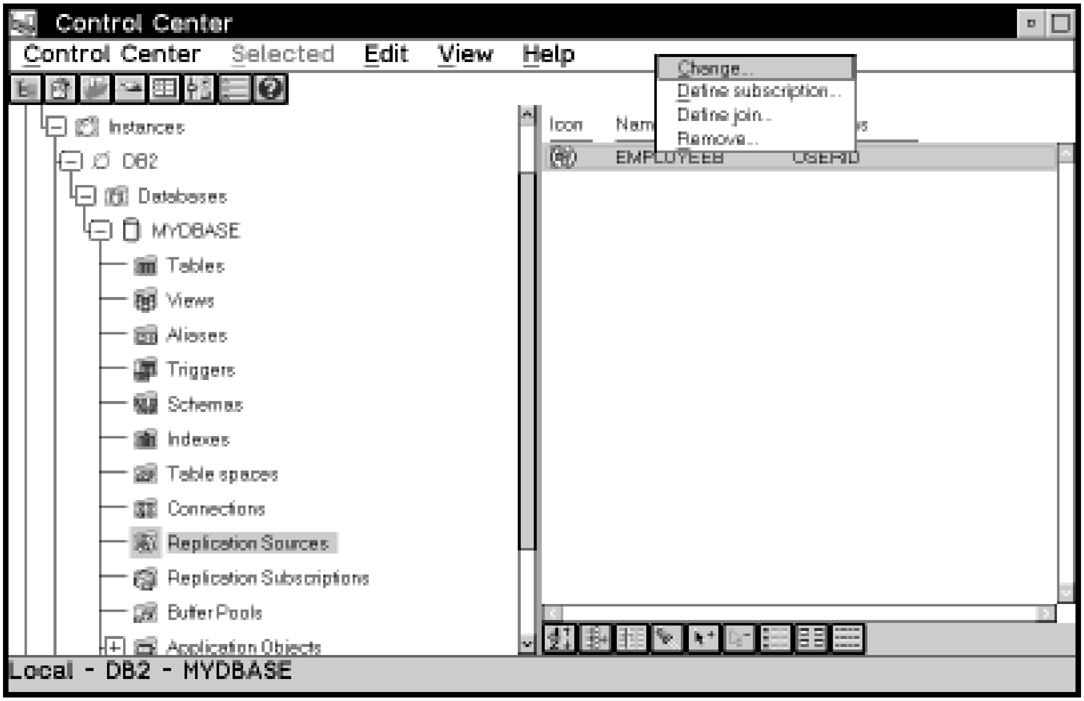
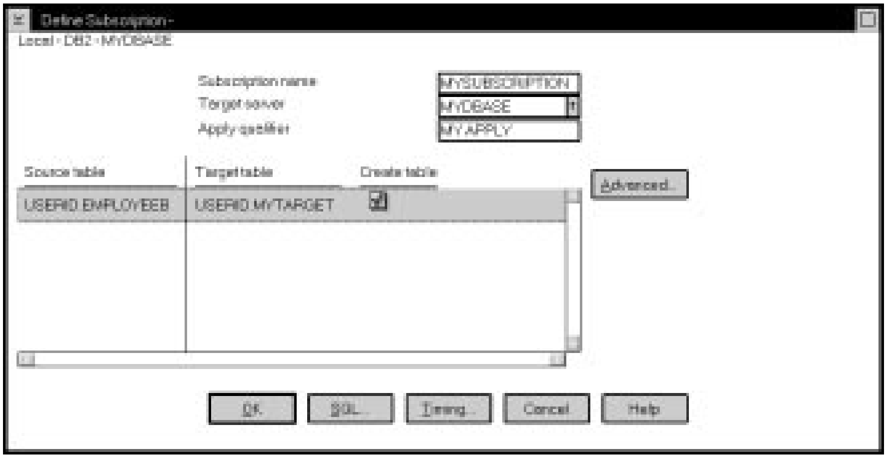


Figure 20. The Replication Sources Folder with Objects and Actions

2. Select one or more replication sources you want to define as sources for the replication subscription, and select **Define subscription** from the pop-up menu. The Define Subscription window opens, as shown in Figure 21.


Define Subscription - Local - DB2 - MYDBASE

Subscription name: MYSUBSCRIPTION
Target server: MYDBASE
Apply qualifier: MYAPPLY

Source table	Target table	Create table
USERID.EMPLOYEEB	USERID.MYTARGET	☑

Figure 21. The Define Subscription Window

3. Type a name for the replication subscription in the **Subscription name** field. The name can be up to 18 bytes in length and can be an ordinary or delimited identifier; for example, `MYSUBSCRIPTION`.

4. Specify the database where the target tables will reside in the **Target server** field. The displayed default is the alias of the source server. This value can be an ordinary or delimited identifier; for example, `MYDBASE`.

5. Specify a character string associating this replication subscription with a unique instance of the Apply program in the **Apply qualifier** field. This value can be an ordinary or delimited identifier and must be in uppercase; for example, `MYAPPLY`.

6. If you want to change the name of the target table, click on and then type over the name in the **Target table** field. You can repeat this step for each target table in the list box. This value can be an ordinary or delimited identifier.

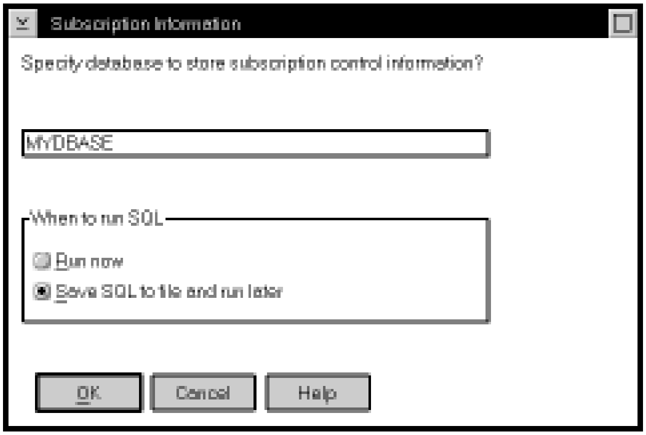
7. Specify whether to create the target table by selecting the **Create table** check box for every target table that you want to create when you subscribe.

   See “Defining a Replication Subscription with a User-Defined Target Table” on page 119 for more information about this option.

8. If you want to customize the target table structure, enhance any data, or specify the timing for the replication subscription, go to “Defining Replication Sources: Advanced Tasks” on page 106.

9. Complete the subscription definition.

   a. Click on **OK** to save the values and close the window. The Subscription Information window opens as shown in Figure 22.



Subscription Information

Specify database to store subscription control information?

MYDBASE

When to run SQL

○ Run now

● Save SQL to file and run later

Figure 22. The Subscription Information Window. Use this window to enter the control server name and specify whether to run now or save the SQL to a file

   b. To specify the control server database, enter the database name in the **Specify database to store subscription control information** field.

   Attention: You can specify an Apply qualifier for the control server while defining the first replication subscription that uses the control server.

This value cannot be changed. See “The Control Server” on page 16 for more information about the Apply qualifier and the control server and how to identify them.

c. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later.

10. After you submit the request, verify that the replication subscription was created by selecting the **Replication Subscription** folder in the object tree and selecting the **Refresh** menu choice.
11. If you defined an event to start the Apply program, populate the event table. See the "Scheduling Subscriptions with the Event Table" section in the Capture and Apply chapter for your platform in this book.
12. To begin replicating data to the target tables, start the Apply program from the command processor window using the name of the control server you specified on the Subscription Information window. See the Capture and Apply chapter for your platform to learn more about the Apply program.

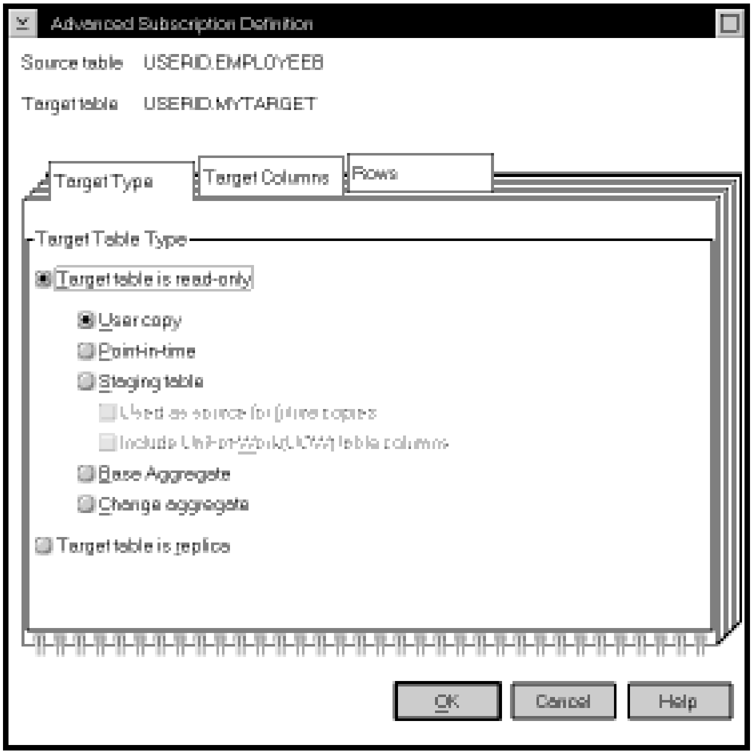
Defining Replication Sources: Advanced Tasks

The following sections describe optional, advanced tasks for defining the replication subscription.

Choosing a Target Table Type

You can specify a specific target table type if you do not want the default target type of user copy. Your application might require a target table that provides a history of changes to the data or an aggregation of columns in the source table. See “Data Transformation Requirements” on page 69 for more information about determining what type of table you should select based on your application's requirements.

Use the Target Type page of the Advanced Subscription Definition notebook to select a target type. Figure 23 on page 107 shows the Target Type page.



Source table USERID.EMPLOYEEB

Target table USERID.MYTARGET

Target Table Type

● Target table is read-only

- ● User copy
- ○ Point-in-time
- ○ Staging table
  - □ Used as source for future copies
  - □ Include Unit-of-Work (UOW) table columns
- ○ Base Aggregate
- ○ Change aggregate

○ Target table is replica

Figure 23. The Target Type Page of the Advanced Subscription Definition Notebook

To specify a target table type:

1. From the Subscription Definition window, select a source and target table combination.
2. Click on **Advanced**. The Advanced Subscription Definition notebook opens.
3. Select one of the following table types:
   - For read-only target tables, you can select:

     User copy
     A target table that matches the source table data exactly at the time of the copy.

     Point in time
     A target table that matches the source table, with a timestamp columns added.

     Base aggregate
     A target table that contains aggregated data for a user table appended at specified intervals.

Change aggregate
A target table that contains aggregated data based on changes recorded for a base table.

Staging
Also known as a CCD table. A staging table that is a join of the source table's change data table and unit-of-work tables. This table can be an internal CCD table or a local or remote CCD table.

- For updateable target tables, select **Target table is replica**, an updateable target table that is used to replicate data back to the origin table or to other target tables.

4. If you are finished using the Advanced Subscription Definition notebook, select **OK** to close the notebook. Otherwise, use the other pages of the notebook to define the target table columns and rows as needed.
5. Complete the subscription. See step 9 on page 105.

Defining the Target Table Structure: Columns and Rows

For some applications, the target table does not need all of the rows or columns that exist in the source table. Or, you might need to rename or create new columns for the target table. For data enhancement, performance, security, or other reasons, you can subset the target table vertically (delete, create new, or rename columns) or horizontally (enter an SQL predicate) using the advanced features of the Define Subscription window.

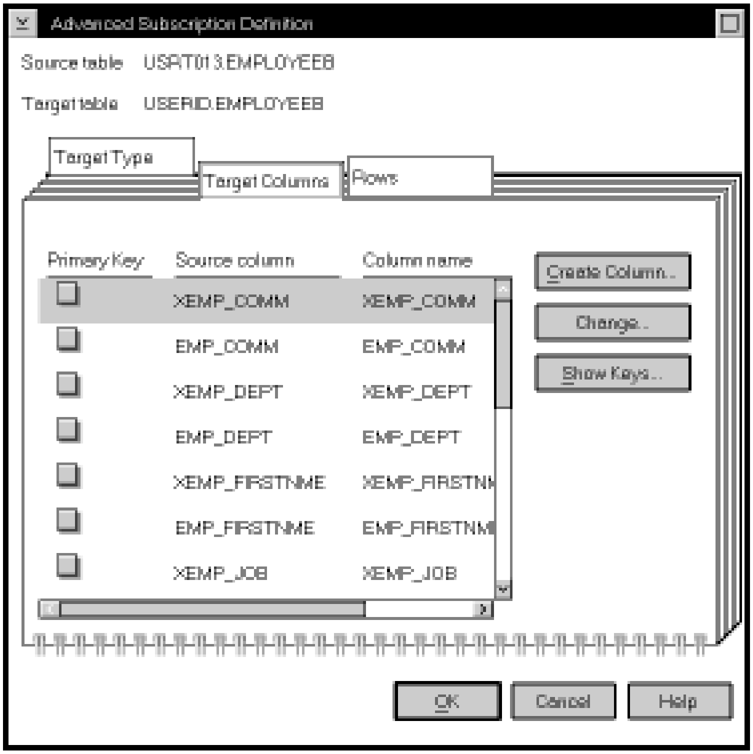
Defining the Target Table Columns

The following section describes how to define the target table columns that are copied to the target table. For data restrictions when defining replication targets, see “Data Restrictions” on page 73.

Attention: Replica target tables must contain the same columns as the source table: they cannot be subsetted, or have columns added or renamed.

To define the target table columns:

1. From the Define Subscription window, select a source and target combination.
2. Click on **Advanced** to open the Advanced Subscription Definition notebook.
3. Open the Target Columns page (Figure 24 on page 109).



Source table USRT013.EMPLOYEEB

Target table USERID.EMPLOYEEB

Primary Key	Source column	Column name
□	XEMP_COMM	XEMP_COMM
□	EMP_COMM	EMP_COMM
□	XEMP_DEPT	XEMP_DEPT
□	EMP_DEPT	EMP_DEPT
□	XEMP_FIRSTNME	XEMP_FIRSTNM
□	EMP_FIRSTNME	EMP_FIRSTNM
□	XEMP_JOB	XEMP_JOB

Figure 24. The Target Columns Page of the Advanced Subscription Definition Notebook. Use this page to subset, create new, and rename columns for the target table.

4. If you want to specify a column as a primary key column for the target table, click on the **Primary Key** check boxes next to the column name.

5. If you want to rename a column, select the column name that you want to edit, and type over the existing column name. This value can be up to 18 single-byte characters and can be an ordinary or delimited identifier.

6. If you want to change a column definition for the target table:

   a. Select **Change**. The Change Column window opens. It has the same fields as the Create Column window, shown in Figure 25 on page 110.

   b. Optional: Change the name of the column in the **Column name** field. The name can be up to 18 single-byte characters. This value can be an ordinary or delimited identifier; for example, `CUST_COUNT`.

   c. Type the SQL expression to change the definition of the column. For example: `COUNT(*)`.

      The expression can contain up to 254 single-byte characters and can be any valid SQL expression. This expression can contain ordinary or delimited identifiers. Columns used in the expression must be valid after-image columns from the source table. These column names are listed in the **Available columns** box.

See your database SQL reference for information on valid SQL expressions. Invalid SQL expressions cause an SQL error when the subscription is processed by the Apply program.

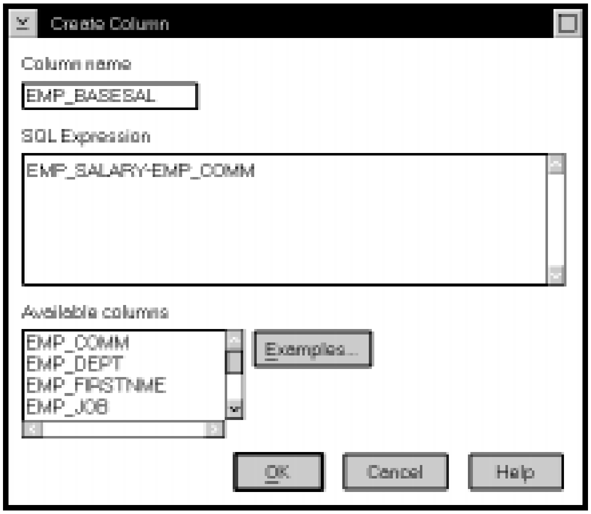
d. Optional: To see examples of SQL expressions, click on **Examples**.

e. Click on **OK** to close the window.

7. If you want to remove a column from the target table, clear the **Subscribe** check box next to the column name.

8. If you want to create a new computed column or use aggregation for the target table:

   a. Click on **Create Column**. The Create Column window opens, as shown in Figure 25.



Column name: EMP_BASESAL

SQL Expression: EMP_SALARY-EMP_COMM

Available columns:
EMP_COMM
EMP_DEPT
EMP_FIRSTNME
EMP_JOB

Figure 25. The Create Column Window. You can define new columns for the target table.

b. Type the name of the column in the **Column name** field. The name can be up to 18 single-byte characters and can be an ordinary or delimited identifier.

c. Type the SQL expression defining the new column.

The expression can contain up to 254 single-byte characters and can be any valid SQL expression and can contain ordinary or delimited identifiers. Columns used in the expression must be valid after-image columns from the source table. These column names are listed in the **Available columns** box.

See your database SQL reference for information on valid SQL expressions. Invalid SQL expressions cause an SQL error when the subscription is processed by the Apply program.

d. Optional: To see examples of SQL expressions, click on **Examples**.

   e. Click on **OK** to close the window.

9. If you are finished using the Advanced Subscription Definition notebook, click on **OK** to close the notebook. Otherwise, use the Rows page to define the target rows as needed.
10. Complete the subscription. See step 9 on page 105.

Defining the Target Table Rows

The following section describes how to define the target table rows that are copied to the target table and the restrictions for doing so.

Row Predicate Restrictions:

- You can use only column names from the **Target columns** list on the Target Columns page; these names are also listed in the **Available columns** list box.
- Do not use before-image columns or computed columns.

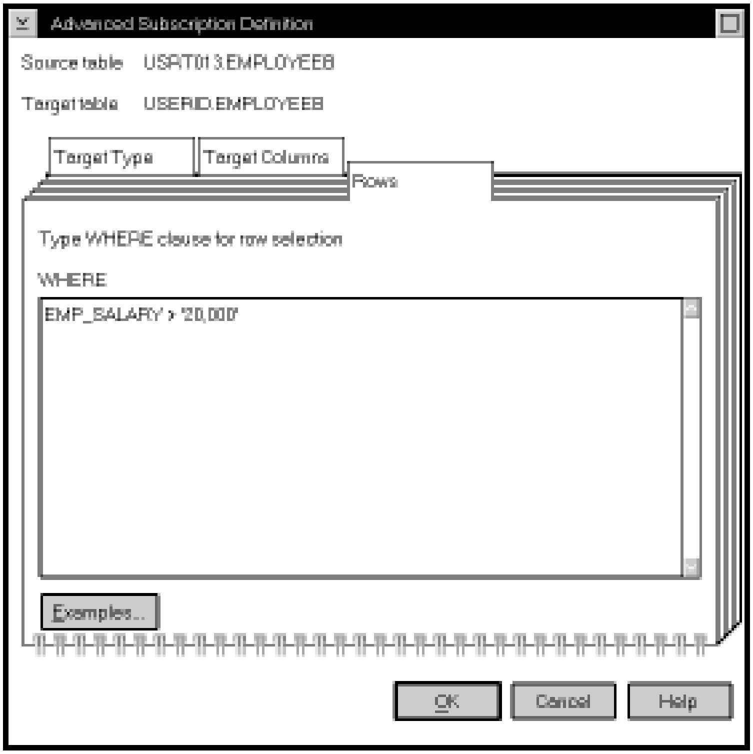
  Before-image columns are supported in change data tables but not user tables. Do not use these columns and predicates without detailed knowledge of the register control table values.
- If you created computed columns on the Target Columns page, you must provide a GROUP BY clause. A computed column is an SQL expression which includes a function that summarizes data. Examples are SUM(WITHDRAWALS), COUNT(*), and AVG(BALANCE). Both base and change aggregate target tables must have a GROUP BY clause.
- Do not type `WHERE` in the clause; it is implied. Type `WHERE` in the clause for subselect statements.
- Do not end the clause with a semicolon (;).
- If your WHERE clause contains the Boolean expression OR, enclose the predicate in parentheses; for example, `(COL1=X OR COL2=Y)`.
- If the target table is a change aggregate table and contains before-image columns, you must include the before-image columns in the GROUP BY clause on the Rows page, even though the before-image columns are not displayed in the **Source columns** box.
- You must provide a dummy WHERE clause when both of the following conditions are true:
  - You are creating an aggregate column that requires a GROUP BY clause.
  - You do not use any other predicate in the **WHERE** field.

  You can receive Apply program run-time errors if you do not provide the dummy WHERE clause in this situation.

To define the target table rows:

1. From the Define Subscription window, select a source and target combination.
2. Click on **Advanced** to open the Advanced Subscription Definition notebook.

3. Open the Rows page (Figure 26 on page 112).



Source table USRT013.EMPLOYEEB

Target table USERID.EMPLOYEEB

Type WHERE clause for row selection

WHERE

EMP_SALARY > '20,000'

Figure 26. The Rows Page of the Advanced Subscription Definition Notebook. You can subset the rows for the target table by typing in a WHERE clause.

4. To specify which rows are copied to the target table, enter an SQL predicate in the **WHERE** field. The predicate can contain ordinary or delimited identifiers. See your database's SQL reference for more information about WHERE clauses.

   Attention: Do not type the word WHERE. It is automatically supplied during processing.

5. To see examples of SQL predicates, click on **Examples**.
6. If you are finished using the Advanced Subscription Definition notebook, click on **OK** to close the notebook.
7. Complete the subscription. See step 9 on page 105.

WHERE clause examples:

The following examples contain WHERE clauses that you can use to filter rows of the target table. These examples are very general and designed for you to use as a model. You can also click on **Examples** on the Rows page to link to additional examples in the online help. See your database's SQL reference for more information about WHERE clauses.

- WHERE clause specifying rows with specific values

  To copy only the rows that contain a specific value, such as MGR for employees that are managers, use a WHERE clause like:

  ```
  EMPLOYEE = 'MGR'
  ```

- WHERE clause specifying rows with a range of values

  To copy only the rows within a range, such as employee numbers between 5000 and 7000 to the target table, use a WHERE clause like:

  ```
  EMPID BETWEEN 5000 AND 7000
  ```

- Dummy WHERE clause

  To support aggregation, assuming that the EMPID column is defined as NOT NULL, use a WHERE clause like the following:

  ```
  EMPID IS NULL
  ```

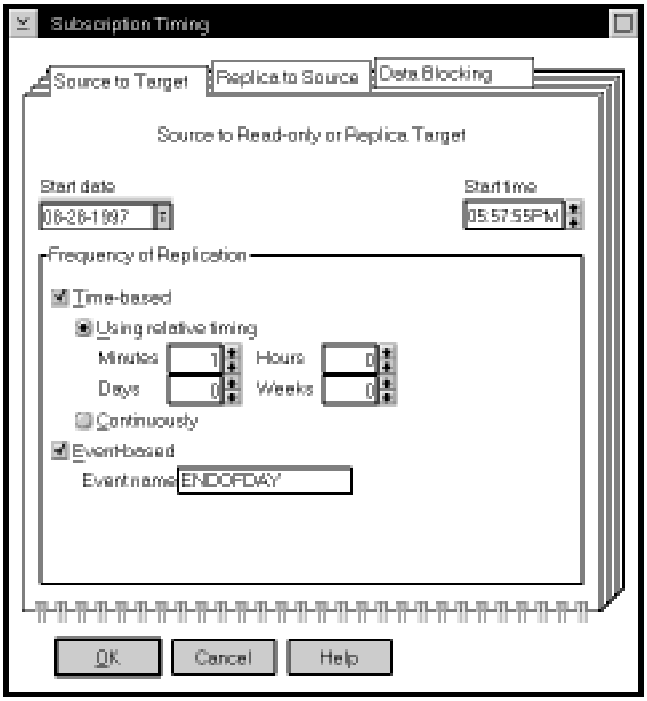
Setting the Copying Schedule: Time or Event Based

You can set the copying schedule when you define a replication subscription. You can use relative or event timing. See “Data Currency Requirements” on page 74 to determine what kind of timing to use.

Set the timing of the replication subscription from the Subscription Timing notebook, accessed by clicking on **Timing** from the Subscription Definition window. The following figure shows the Subscription Timing notebook. There are two timing pages. Both pages have same fields.

Source to Target
The timing information for replicating changed data from the replication source tables to the target tables.

Replica to Source
The timing information for replicating changed data from replica tables to the replication source tables. Complete this page only for replication subscriptions with replica tables.



Source to Read-only or Replica Target

Start date: 08-28-1997

Start time: 05:57:55PM

Frequency of Replication

☑ Time-based

● Using relative timing

Minutes: 1 Hours: 0

Days: 0 Weeks: 0

○ Continuously

☑ Event-based

Event name: ENDOFDAY

Figure 27. The Subscription Timing Notebook. You can specify relative or event timing for the replication subscription.

To specify a time-based frequency:

1. From the Define Subscription window, click on **Timing**. The Source to Target page of the Subscription Timing notebook opens.
2. Specify the date on which you want the Apply program to start replicating the replication subscription in the **Start date** field.
3. Specify the time that you want the replication to start in the **Start time** field by typing over the existing characters or using the spin buttons.
4. Select the **Time-based** check box.
5. Choose the type of interval that you want by selecting one of the following radio buttons:

   Using relative timing Specifies a specific interval, in minutes, hours, days, or weeks. Use the radio dials to specify the interval that you want.

   Continuously Specifies to replicate continuously.
6. If you are finished with this notebook, click on **OK**. Otherwise, click on one of the other notebook tabs to specify replica to source timing or data blocking for the replication subscription.
7. Complete the subscription. See step 9 on page 105.

To specify an event-based frequency:

1. From the Define Subscription window, click on **Timing**. The Source to Target page of the Subscription Timing notebook opens.
2. Click on the **Event-based** check box.
3. In the **Event** field, type the event name with which you will populate the event table to trigger replication. This value can be an ordinary or delimited identifier.
4. If you are finished with this notebook, click on **OK** to close the notebook. Otherwise, click on one of the other notebook tabs to specify replica to source timing or data blocking information for the replication subscription.
5. Complete the subscription. See step 9 on page 105.

See the "Scheduling Subscriptions with the Event Table" section in the Capture and Apply chapter for your platform in this book.

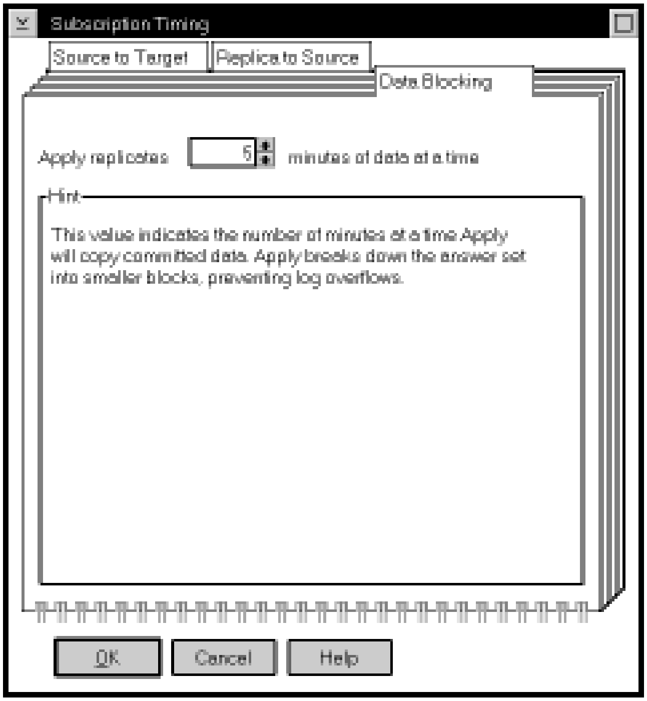
Specifying Mini-Cycles for the Apply Program to Copy Committed Data

You can specify the number of minutes of data that the Apply program copies at a time. This value helps the Apply program break down a large block of changed data into smaller subscription cycles, preventing spill file or log overflows. See “Data Blocking: Using MAX_SYNCH_MINUTES for Large Jobs” on page 22 for information about data blocking and “Data Blocking for Large Volumes of Changes” on page 61 for information on how to determine this value.

To specify a data blocking value:

1. From the Define Subscription window, click on **Timing**. The Subscription Timing notebook opens.
2. Open the Data Blocking page as shown in Figure 28 on page 116.

You can specify the number of minutes of data copied at a time.



Apply replicates 5 minutes of data at a time

This value indicates the number of minutes at a time Apply will copy committed data. Apply breaks down the answer set into smaller blocks, preventing log overflows.

Figure 28. The Data Blocking Page of the Subscription Timing Notebook

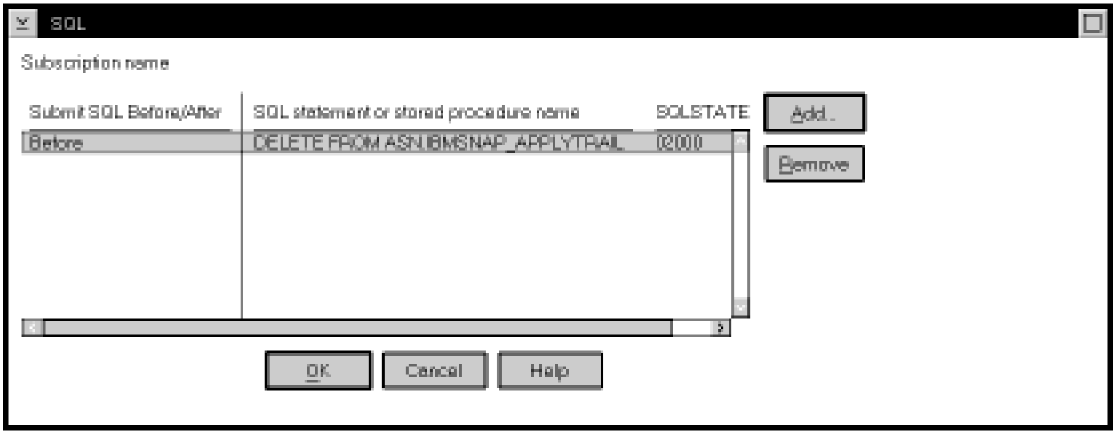
3. Enter the interval value of the number of minutes that you want the Apply program to copy data at a time.

4. If you are finished using the Advanced Subscription notebook, click on **OK** to close the notebook.

5. Complete the subscription. See step 9 on page 105.

Defining SQL Statements or CALL Procedures for the Replication Subscription

You can define SQL statements or CALL procedures to be run before or after the Apply program copies the data from the source to the target table.



Submit SQL Before/After	SQL statement or stored procedure name	SQLSTATE
Before	DELETE FROM ASN.IBMSNAP_APPLYTRAIL	02000

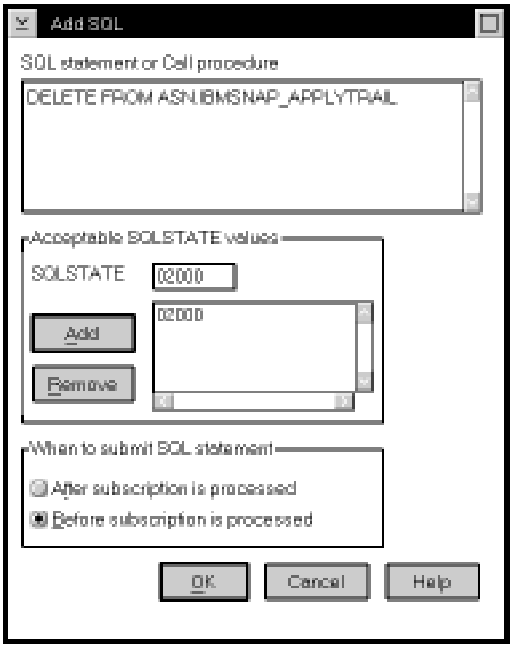
Figure 29. The SQL Window. You can add or delete SQL or CALL procedures for processing before and after the replication subscription.

To specify SQL statements or CALL procedures statements for the replication subscription:

1. From the Define Subscription window, click on **SQL**. The SQL window opens (Figure 29).

   Use the SQL window to add or remove SQL statements or CALL procedures that are submitted either before or after the replication subscription is processed. The statements are processed in the order they appear in the list.

2. Click on **Add**. The Add SQL window opens, as shown in Figure 30 on page 118.



SQL statement or Call procedure: DELETE FROM ASN.IBMSNAP_APPLYTRAIL

Acceptable SQLSTATE values

SQLSTATE: 02000

02000

When to submit SQL statement

○ After subscription is processed

● Before subscription is processed

Figure 30. The Add SQL Window. You can add SQL statements or CALL procedures.

3. Type the SQL statement or stored procedure name in the **SQL statement or Call procedure** field. The stored procedure name must begin with `CALL`. This field can contain ordinary or delimited identifiers.
4. If you need to enter a valid SQLSTATE that the Apply program needs to pass, type a valid 5-byte SQLSTATE value in the **SQLSTATE** field and click on **Add**. The value is added to the **Acceptable SQLSTATE values** box. You can enter up to 10 values. The Control Center interprets these values as successful execution.
5. Specify whether you want to submit the SQL statement or CALL procedure before or after the replication subscription is processed by clicking on the appropriate radio button in the **When to submit SQL statement** field.
6. Click on **OK**. The statement is added to the box in the SQL window and the Add SQL window closes.
7. Repeat for each SQL statement or CALL procedure.
8. Click on **OK** to return to the Define Subscription window.
9. Complete the subscription. See 9 on page 105.

To remove an SQL statement for CALL procedure:

1. From the Define Subscription window, click on **SQL**. The SQL window opens.
2. Select the statement you want to remove.
3. Click on **Remove**.

4. Click on **OK** to close the window.
5. Complete the subscription. See step 9 on page 105.

Defining a Replication Subscription with a User-Defined Target Table

The Control Center allows you to use a previously defined DB2 table as the target table in a replication subscription. This type of target table is known as a user-defined target table. With this feature, you can define a replication subscription with a target table that is defined outside of replication administration in the Control Center.

The following sections explain how to define a user-defined target table, the restrictions, and maintaining the target table.

To define a subscription with a user-defined target table:

1. Refer to Chapter 20, “Table Structures” on page 265 to determine the structure for the target table type. For example, if you are defining a subscription for a base aggregate target table, refer to the table structure definition for base aggregate target tables.
2. Alter the target table to add any required columns, such as timestamp columns.
3. Create a unique index for condensed tables (point-in-time or condensed CCD tables). Define the index using the SQL statements provided in Chapter 20, “Table Structures” on page 265.
4. Define the subscription to match the user-defined target table structure. This task includes creating new columns, subsetting columns, changing column names, or renaming before-image columns to include the before-image column prefix.

   On the Define Subscription window:

   a. Clear the **Create table** check boxes for the table names for which you are providing the target tables.
   b. Type the user-defined target table name in the **Target table** field.
   c. If you want to subset columns or rows, enhance data, or specify a target table type other than user copy, click on **Advanced** to open the Advanced Subscription Definition notebook.
      1) If you want to select an alternate table type, see “Choosing a Target Table Type” on page 106.
      2) If you want to modify the target table columns to match the user-defined target table, see “Defining the Target Table Structure: Columns and Rows” on page 108.
      3) If you want to subselect the rows or use an aggregate expression, see “Defining the Target Table Structure: Columns and Rows” on page 108.

Restrictions

The following conditions cause run-time errors:

- The subscription definition contains fewer columns than exist in the user-defined target table.
- The new columns in the subscription definition do not allow nulls or have defined default values.
- The target table and unique index do not exist when you run the Apply program.

Maintaining a User-Defined Target Table

When you use a user-defined target table during subscription, you assume the responsibilities listed below. The Control Center does not check for inconsistencies between the subscription definition and a user-defined target table. You must complete the following activities:

- Ensure that a target table and unique index (for point-in-time and condensed CCD tables) exist that match the subscription definition when running the Apply program.
- Debug any inconsistencies that exist between the target table and the subscription definition.
- For a user-defined CCD target table, define the subscription to match the attributes specified for the target table by specifying:
  - The target table type as a staging table
  - Whether the table has one or multiple changes per key

  For CCD target tables: When a subscription is to be defined on a base table having a user-defined internal CCD, ensure that the internal CCD table was created to match the subscription definition.

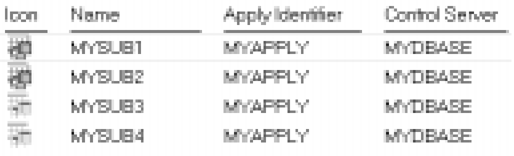
  Because an internal CCD table replaces the join of the CD and UOW tables as the source for updates to the target table, ensure that the CCD table exists before attempting to subscribe to the base table again. If the internal CCD target table is not created, the source table will not be available for another subscription to the base table.

Activating and Deactivating Replication Subscriptions

You can control the active status of a replication subscription. This feature is useful when you want to temporarily deactivate a replication subscription without removing it. When you deactivate a replication subscription, the Apply program completes its current processing cycle and then stops.

To deactivate a replication subscription:

1. Select the **Replication Subscriptions** folder from the object tree. The replication subscriptions appear in the contents pane.
2. Verify that the replication subscription is active by looking at the status icon to the left of the replication subscription object that you want to deactivate. Figure 31 on page 121 shows activated and deactivated replication subscriptions.



Icon	Name	Apply Identifier	Control Server
	MYSUB1	MYAPPLY	MYDBASE
	MYSUB2	MYAPPLY	MYDBASE
	MYSUB3	MYAPPLY	MYDBASE
	MYSUB4	MYAPPLY	MYDBASE

Figure 31. The active and inactive icons. The active and inactive icons show which replication subscriptions have been activated and deactivated.

The first two are active, the second two are inactive.

If the icon for the replication subscription that you are working with shows that the replication subscription is active, continue to the next step. If the icon is inactive, do not continue with this task because the replication subscription is already inactive.

3. In the contents pane, select the replication subscription object that you want to deactivate and select **Activate/Deactivate** from the pop-up menu.

The replication subscription is deactivated, and the status icon is updated to reflect the change.

To reactivate a deactivated replication subscription:

1. Select the **Replication Subscriptions** folder from the object tree. The replication subscriptions appear in the contents pane.
2. Verify that the replication subscription is inactive by looking at the status icon to the left of the replication subscription object that you want to activate. If the icon shows that the replication subscription is inactive, continue to the next step. If the icon is active, do not continue with this task because the replication subscription is already active.
3. In the contents pane, select the replication subscription object that you want to reactivate and select **Activate/Deactivate** from the pop-up menu.

The replication subscription is reactivated and the status icon is updated to reflect the change.

Cloning a Replication Subscription to Another Server

You can clone a replication subscription to another server. Cloning creates an exact copy of an existing replication subscription on a different target server, using a different Apply qualifier. You can clone one or more replication subscriptions at a time. The Control Center updates the control tables at the control server.

To clone a replication subscription:

1. Select the **Replication Subscriptions** folder from the object tree. The replication subscriptions appear in the contents pane.
2. In the contents pane, select the replication subscription object that you want to modify and select **Clone** from the pop-up menu. The Clone window opens.

3. Specify a new Apply qualifier in the **Apply qualifier** field. This value can be an ordinary or delimited identifier.
4. Specify the new target server in the **Target server** field. This value can be an ordinary or delimited identifier.
5. Click on **OK** to save the values and close the window. The Subscription Information window opens.
6. To specify the control server database, enter the database name in the **Specify database to store subscription control information** field.
7. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.
8. After you submit the request, verify that the replication subscription was cloned by selecting the **Replication Subscription** folder in the object tree and selecting the **Refresh** menu choice. Check the Apply qualifier and target server name for the replication subscription.

Viewing or Changing an Existing Replication Subscription

You can change a subset of the replication subscription values, primarily those that do not affect the structure of the target tables.

To change the replication subscription:

1. Select the **Replication Subscriptions** folder from the object tree. The replication subscriptions appear in the contents pane.
2. In the contents pane, select the replication subscription object that you want to modify and select **Change** from the pop-up menu. The Change Subscription Definition window opens.
3. You can change the following values in the Change Subscription window and subwindows:
   - The Row predicate in the Advanced Subscription notebook. The new predicate is not applied to existing rows in the target table. The predicate is used starting with the next subscription cycle for the replication subscription. See “Defining the Target Table Rows” on page 111 for more information and examples.
   - The SQL or CALL procedure for before or after copying in the SQL window. See “Defining SQL Statements or CALL Procedures for the Replication Subscription” on page 116 for more information.
   - The timing values in the Subscription Timing notebook. See “Setting the Copying Schedule: Time or Event Based” on page 113 for more information.
   - The data blocking value in the Subscription Timing notebook. See “Specifying Mini-Cycles for the Apply Program to Copy Committed Data” on page 115 for more information.

4. Click on **OK** to save the values and close the window. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.

Removing Replication Subscriptions

When you no longer need a replication subscription definition, or the requirements have changed enough that you need to redefine it, you can remove it. Removing a replication subscription definition deletes information about it from the control tables and deletes the target table from the target server. If you have selected to drop table spaces in the Tools Settings notebook, the table space is also dropped.

You can remove replication subscription definitions that are dependent on replication sources that you are removing.

To remove a replication subscription:

1. Select one or more replication subscription objects from the contents pane and select **Remove** from the pop-up menu. The Remove Subscription Confirmation window opens.
2. Click on **Yes** to save the values and close the window. The Run Now or Save SQL window opens. Specify whether to submit the replication request now or save the SQL statements generated by the action in a file to be run later. See “Customizing and Running Replication SQL Files” on page 89 to learn how to submit the request now or later.
3. After you submit the request, verify that the replication subscription was removed by selecting the **Replication Subscriptions** folder in the object tree and selecting the **Refresh** menu choice.

Part 4. IBM Replication Capture and Apply

This part describes how to configure, operate and troubleshoot the Capture and Apply programs. Each chapter describes the Capture and Apply programs for a particular platform.

Chapter 10. Capture and Apply for MVS

This chapter describes how to set up, operate, and troubleshoot Capture and Apply for MVS.

Read the following sections before reading the sections on Operating Capture for MVS:

- “Setting Up the Capture and Apply Programs”
- “Specifying Tuning Parameters for the Capture Program”
- “Restrictions When Running the Capture Program” on page 128
- “Authorization for Running the Capture Program” on page 129
- “Recovering from Severe Errors” on page 129

Setting Up the Capture and Apply Programs

Setting up consists of installing the Capture and Apply programs and configuring the source, target, and control servers. Capture for MVS and Apply for MVS are packaged in SMP/E format. The installation sequence for each program consists of:

- Using SMP/E to install
- Providing APF authorization
- Creating and loading the messages file
- Binding to the DB2 subsystem
- Customizing invocation JCL

Capture for MVS uses the DB2 for MVS/ESA Instrumetation Facility Interface (IFI) to retrieve log records. Make sure to apply the correct DB2 maintenance before installing the Capture program.

See "Installation Steps for Capture for MVS" in the Capture for MVS program directory for instructions about installing the Capture for MVS program. See "Installation Steps for Apply for MVS" in the Apply for MVS program directory for instructions about installing the Apply for MVS program. System programmers should read all of the steps in the program directory through step 9, "Installation Checklist." System administrators should read step 10, "Update and Execute the Bind Job," and step 11, "Starting the Run Job" in the program directory.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit
The number of minutes to keep the change data table rows and the unit-of-work (UOW) table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit

The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If Capture shuts itself down, you should perform a cold start if the database does not have or support the archive log. DB2 for MVS 3.1 does not support the archive log.

Commit interval

The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if the Capture and Apply programs are running at the same time. This precaution helps to avoid locking overhead for Capture running on DB2 for MVS 3.1 where uncommitted read is not supported. If the Apply program is *not* running at the same time as the Capture program, you can set the commit interval no higher than the DB2 timeout interval.

Prune interval

The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is ignored if you start Capture with the NOPRUNE option; however, you can override this option with the PRUNE command.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

```
UPDATE TABLE ASN.IBMSNAP_CCPPARMS
SET RETENTION_LIMIT=number_of_minutes,
LAG_LIMIT=number_of_minutes,
COMMIT_INTERVAL=number_of_seconds,
PRUNE_INTERVAL=number_of_seconds,
```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.
- Drop a replication source table.

- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Capture for MVS does not support certain programmatic statements for the source table. See “Data Restrictions” on page 73 for a list of these statements.

Other Capture program restrictions are:

- Capture for MVS does not support compressed data for a source table if you use DB2 3.1 for MVS/ESA.
- User tables must be created or altered using the DATA CAPTURE CHANGES option.
- Capture does not capture any changes made by DB2 utilities.
- If you are using DB2 for MVS/ESA Version 4.1 or later, only one instance of the Capture program can run on any subsystem that is a member of a data-sharing group. See “Capture for MVS in a Data-Sharing Environment” on page 14 for more information about data sharing.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have the following authorizations:

- SELECT, UPDATE, INSERT, and DELETE privilege for all Capture-related tables created explicitly, and any Capture-related tables the Control Center implicitly creates
- SELECT privilege for the DB2 catalog table, SYSIBM.SYSTABLES
- TRACE privilege
- MONITOR1 and MONITOR2 privilege
- EXECUTE privilege for the Capture plan

Recovering from Severe Errors

Capture for MVS provides you with the following tools to assist you if a severe error occurs:

- Alert generation
- Trace buffer
- Trace output
- Storage dump

The following sections provide information on these tools.

Alert Generation

If a severe error occurs, Capture for MVS alerts NetView if NetView is active. The alert uses the NMVT format for a generic alert defined by the SNA generic alert architecture. If NetView is unavailable, diagnostic information is still available because Capture for MVS error messages are sent to the MVS console.

Trace Buffer

Capture for MVS puts a small amount of critical diagnostic data in a wraparound trace buffer during processing. Each trace buffer entry describes current data capture status. If a severe error occurs, Capture prints the trace buffer before termination. The printing of the trace buffer supplements the Capture program error message.

Trace Output

When an error occurs, you can run the Capture program with the TRACE option. When you use this option, IBM replication writes trace information logic flow to STDOUT.

Storage Dump

When Capture for MVS terminates with a severe error, it saves critical diagnostic data in the CEEDUMP data set. This information is more detailed than the information in the Capture program trace buffer.

Operating Capture for MVS

The replication administrator can use the commands in this section to perform the following Capture for MVS tasks:

- Starting
- Scheduling
- Stopping
- Suspending
- Resuming
- Reinitializing
- Pruning

You can submit the commands from TSO or the MVS console.

Before You Start the Capture Program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where Capture is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS

  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC

  You can also create these control tables manually by running the DPCNTL.* file from the RUN SQL Files window.

- Bind the Capture program to the source server from which the Capture program will capture changes. Refer to the Capture for MVS program directory for information on running the bind program for Capture.
- Ensure that ASN.IBMSNAP_REGISTER has at least one entry in it by defining a replication source with the Data Capture Changes option.

Starting Capture for MVS

After you start the Capture program, it runs continuously until you stop it or it detects an error.

To start the Capture program:

1. Prepare the JCL for MVS by specifying the appropriate optional invocation parameters in the PARM field of the ASNLRN*nn* DD statement. Tailor the JCL to meet your site's requirements. Sample JCL appears in the program directory for Capture for MVS.

   An example of this line in the invocation JCL is:

   ```
   //ASNLRNnn EXEC PGM=ASNLRPnn,PARM='DB2_subsys_name NOTERM WARMNS NOPRUNE'
   ```

   where *nn* is the level of DB2 for MVS.

   Table 9 defines the optional parameters.

2. Submit the JCL from TSO or from the MVS console. Capture for MVS can run either as a batch job or a started task.

Table 9 defines the parameters.

Table 9 (Page 1 of 2). ASNLRP Invocation Parameter Definitions for MVS

Parameter	**Definition**
DB2_*subsys_name*	The default is DSN for the subsystem name. This must be the first parameter.
TERM (default)	Terminates the Capture program if DB2 is terminated.
NOTERM	Keeps the Capture program running if DB2 is terminated with MODE(QUIESCE). When DB2 comes up, Capture starts in WARM mode and begins capturing where it left off when DB2 terminated. If DB2 terminates via FORCE or due to abnormal termination, the Capture program terminates even if you selected this parameter.

Table 9 (Page 2 of 2). ASNLRP Invocation Parameter Definitions for MVS

Parameter	Definition
WARM (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. Refer to “Warm and Cold Starts” on page 134 for more information.
WARMNS	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With WARMNS, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, Capture terminates instead of switching to a cold start as when WARM is specified. Refer to “Warm and Cold Starts” on page 134 for more information.
COLD	The Capture program starts up by deleting all rows in its change data table, UOW table, pruning control table, and trace table during initialization. Refer to “Warm and Cold Starts” on page 134 for more information.
PRUNE (default)	The Capture program automatically prunes the change data and the UOW tables that the Apply program has copied since the last pruning.
NOPRUNE	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the PRUNE command. See the PRUNE command for more information.
NOTRACE (default)	No trace information is written.
TRACE	Writes debug trace messages to the standard output, *stdout.*

Scheduling Capture for MVS

Use either the $TA JES2 command or the AT NetView command to start Capture for MVS at a specific time. You must:

1. Create a procedure that calls Capture for MVS in the PROCLIB.
2. Modify the ICHRIN03 RACF module (or appropriate definitions for your MVS security package) to associate the procedure with a user ID.
3. Link edit the module in SYS1.LPALIB.

Stopping Capture for MVS

```
►►──F──jobname,STOP──────────────────────────────────────────►◄
```

Purpose

Use the STOP command to stop the Capture program in an orderly way and commit the log records that it processed up to that point.

Usage

Enter this command from TSO or the MVS console before:

- Removing an existing replication source
- Modifying an existing replication source

Suspending Capture for MVS

```
►►──F──jobname,SUSPEND──────────────────────────────────────────►◄
```

Purpose

Use the SUSPEND command to relinquish MVS resources to operational transactions during peak periods without damaging the Capture program environment.

This command suspends the Capture program until you issue the RESUME command.

Usage

Enter the SUSPEND command from TSO or the MVS console.

Attention: Do not use SUSPEND when canceling a replication source. Instead, stop the Capture program by entering the STOP command.

Resuming Capture for MVS

```
►►──F──jobname,RESUME───────────────────────────────────────────►◄
```

Purpose

Use the RESUME command to restart the Capture program if you suspended it using the SUSPEND command.

Usage

Enter the RESUME command from TSO or the MVS console.

Reinitializing Capture for MVS

```
►►──F──jobname,REINIT───────────────────────────────────────────►◄
```

Purpose

Use the REINIT command to have the Capture program reread the ASN.IBMSNAP_REGISTER tables or the ASN.IBMSNAP_CCPPARMS table while it is running. Use this command when you add a new replication source with Control Center while the Capture program is running.

This command ensures that the Capture program recognizes the new replication source. It also rereads the ASN.IBMSNAP_CCPPARMS table for any changes made to the tuning parameters.

Attention: Do not use REINIT to reinitialize the Capture program after canceling a replication source or dropping a replication source table while the Capture program is running. Instead, stop the Capture program and restart it using the WARM option.

Usage

Enter the REINIT command from TSO or the MVS console.

Pruning the Change Data and Unit-of-Work Tables

```
►►──F──jobname,PRUNE─────────────────────────────────────────────────►◄
```

Purpose

Use the PRUNE command to start pruning the change data (CD) table and UOW table if you used the NOPRUNE invocation parameter to disable pruning when you started the Capture program.

This command prunes the CD and UOW tables once.

Usage

Enter the PRUNE command from TSO or the MVS console. The Capture program issues the message ASN0124I when the command is successfully queued.

During pruning, if you stop or suspend Capture, pruning does not resume after you enter the RESUME command. You must enter the PRUNE command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──F──jobname,GETLSEQ───────────────────────────────────────────────►◄
```

Purpose

Use the GETLSEQ command to provide the current log sequence number and the timestamp. You can use this number to determine how far Capture has read the DB2 log.

Usage

Enter the GETLSEQ command from TSO or the MVS console. The Capture program issues the message ASN0125I indicating when the current log sequence number is successfully processed.

Warm and Cold Starts

This section explains how Capture handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the warm start table, ASN.IBMSNAP_WARM_START, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, UOW table, or change data tables.

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, an operator might cancel the Capture program or stop DB2. In this case, the Capture program uses the change data, UOW, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (The Apply program; performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued. For information about handling gap messages, see “Problems Using the Apply Program” on page 143 and read the portion about forcing the Apply program to perform a full refresh.)

Automatic Cold Starts

Sometimes the Capture program automatically switches to a cold start, even if you specified a warm start (not WARMNS). For example:

- The warm start log sequence lags behind the current log sequence by more than the LAG_LIMIT value (as specified in the tuning parameters table ASN.IBMSNAP_CCPPARMS).
- The warm start log sequence is not available on the DB2 active log. (This situation applies only for DB2 for MVS 3.1.)
- You invoke the Capture program for the first time.

  The first time that you invoke the Capture program, you see message ASN0102W, indicating that the warm start failed. Capture switches to a cold start. You can ignore this message when first invoking the Capture program.

For each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, or if someone brings down the DB2 tablespace containing the change data table. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Operating Apply for MVS

A replication administrator can use the commands in the following sections to perform the following Apply for MVS tasks:

- Starting
- Scheduling
- Stopping

You can submit the commands from TSO or an MVS console.

Before You Start the Apply Program

Before you start the Apply program, ensure that:

- The administrator defined the control tables.
- At least one subscription is created and activated.
- The Apply program package is created.

  See the Apply for MVS program directory for information on BIND programs to create the Apply program packages. You must bind the Apply program to both the source and target databases.
- You have the proper authorization. See "Authorization Requirements for the Capture Program" on page 66 for information about authorization for the Apply program.

Starting Apply for MVS

After you start Apply, it runs continuously until:

- You stop it in an orderly way.
- An unexpected error or failure occurs.

To start Apply for MVS:

1. Prepare the JCL for MVS by specifying the appropriate optional invocation parameters in the PARM field of the ASNARUN*nn* command. Tailor the JCL to meet your site's requirements. Sample JCL appears in the program directory for Apply for MVS.

   An example of this line in the invocation JCL is:

   ```
   //ASNARUN EXEC PGM=ASNAP,PARM='Apply_qual DB2_subsystem_name DISK'
   ```

   Table 10 on page 137 defines the required and optional positional parameters.

Table 10. ASNARUN Invocation Parameter Definitions

Parameter	Definition
Apply_qual	Specifies the Apply program qualifier the Apply program instance uses to identify the subscriptions to be served. This is a positional parameter that is required as the first parameter.
DB2_subsystem_name	The default is DSN. This is a positional parameter that is required as the second parameter.
Control_server_name	Specifies the name of the server where the replication control tables will reside. If you do not specify this optional positional parameter, the default is the current server.
LOADXit	Specifies that the Apply program is to invoke ASNLOAD, an exit program that can call an IBM or multivendor utility, to initialize a target table. This keyword parameter is optional.
NOLOADXit (default)	Specifies that the Apply program will not invoke ASNLOAD. This keyword parameter is optional. Currently, no utility program is available for use by ASNLOAD on DB2 for MVS.
MEMory (default)	Specifies that a memory file stores the fetched answer set. The Apply program fails if there is insufficient memory for the answer set. This keyword parameter is optional.
DISK	Specifies that a disk file stores the fetched answer set. This keyword parameter is optional.
INAMsg (default)	Specifies that the Apply program is to issue a message when the Apply program is inactive.
NOINAMsg	Specifies that the Apply program will not issue this message. This keyword parameter is optional.
NOTRC (default)	Specifies that the Apply program does not generate a trace.
TRCERR	Specifies that the Apply program generates a trace that contains only error information. This keyword parameter is optional.
TRCFLOW	Specifies that the Apply program generates a trace that contains error and execution flow information. This keyword parameter is optional.
NOTIFY	Specifies that the Apply program is to invoke ASNDONE, an exit program that returns control to the user when the Apply program processing ends. This keyword parameter is optional.
NONOTIFY (default)	Specifies that the Apply program will not invoke ASNDONE.

Scheduling Apply for MVS

Use either the $TA JES2 command or the AT NetView command to start Apply for MVS at a specific time.

To start Apply for MVS at a specific time:

1. Create a procedure that calls Apply for MVS in the PROCLIB.
2. Modify the ICHRIN03 RACF module (or appropriate definitions for your MVS security package) to associate the procedure with a user ID.

3. Link edit the module in SYS1.LPALIB.

See *MVS/ESA JES2 Commands* for more information about using the $TA JES2 command, and See *NetView for MVS Command Reference* for more information about using the AT NetView command.

Stopping Apply for MVS

Enter the following command from TSO or an MVS console:

```
►►──P──jobname──────────────────────────────────────────────────────────►◄
```

Additional Apply Program Operations

The following sections provide information about performing the additional Apply program functions of scheduling copies and initiating a forward recovery.

Scheduling Subscriptions with the Event Table

You can include an event name in your replication subscription. Your subscription cycles can run according to a timer, an event occurrence, or both. See “Setting the Copying Schedule: Time or Event Based” on page 113 for more information about the purpose and usage of event scheduling.

You can schedule a subscription by manipulating the event table. The DDL statement for the event table appears as follows:

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_EVENT (
  EVENT_NAME CHAR(18) NOT NULL,
  EVENT_TIME TIMESTAMP NOT NULL,
  END_OF_PERIOD TIMESTAMP);
```

For example, to start the Apply program at 6:00 p.m. on 9/24/97 and copy changes from the source server up to 4:00 p.m. on the same date, you would insert the following values in ASN.IBMSNAP_SUBS_EVENT:

```
('EVENTA','1997-09-24-18.00.00.00000','1997-09-24-16.00.00.00000'),
```

A subscription becomes eligible for processing when EVENT_TIME falls between LASTSUCCESS and the control server's CURRENT TIMESTAMP. The END_OF_PERIOD column enables you to specify a timestamp upper limit for change data.

Loading Large Copies

As an alternative to using the LOADX option when running the Apply program program, you can automate the loading of many DB2 copies by using the following technique to perform your own full refresh on behalf of Apply for MVS. By doing so, Apply for MVS assumes that *it* has initialized your large point-in-time copies. You must use this technique to fully refresh tables with referential integrity constraints in order to bypass referential integrity checking.

The following example assumes that you want to perform a load of the tables within the replication subscription SET001 with an Apply program qualifier of APPLY001.

1. Optional: Disable the full refresh capability for the applicable source tables as shown in the following SQL statements: the Apply program then issues an error message, rather than performing a full refresh if this procedure is not performed correctly.

   ```
   UPDATE ASN.IBMSNAP_REGISTER SET DISABLE_REFRESH = 1
   WHERE SOURCE_OWNER = 'source owner' AND
   SOURCE_TABLE = 'source table'
   ```

2. Ensure that the Apply program is inactive, or deactivate the applicable replication subscription as shown in the following SQL statements:

   ```
   UPDATE ASN.IBMSNAP_SUBS_SET SET ACTIVATE = 0
   WHERE SET_NAME = 'SET001' AND APPLY_QUAL = 'APPLY001'
   ```

3. Update SYNCHPOINT and SYNCHTIME in ASN.IBMSNAP_PRUNCNTL for each row that corresponds to a member of the replication subscription as shown in the following SQL statements:

   ```
   UPDATE ASN.IBMSNAP_PRUNCNTL
   SET SYNCHPOINT = x'00000000000000000000',
   SYNCHTIME  = CURRENT TIMESTAMP
   WHERE SET_NAME = 'SET001' AND APPLY_QUAL = 'APPLY001'
   ```

4. Verify that Capture has processed the updates in the step above.

   - If Capture has not started capturing changes for these tables, this update initiates the process. ASN.IBMSNAP_TRACE will contain GOCAPT messages for the source tables.

   - If Capture has previously started capturing changes for these tables (other subscriptions exist for the tables and these have already been activated) you can verify that this pruning control table update was successful by querying the same SYNCHPOINT values that were updated to hex zeroes. When these values no longer appear as zeroes, Capture has performed the necessary processing for you to continue.

5. Unload the source table and load the target table, using DB2 or vendor utility programs.

6. Update ASN.IBMSNAP_SUBS_SET as shown in the following SQL statements:

   ```
   UPDATE ASN.IBMSNAP_SUBS_SET
   SET LASTRUN = CURRENT TIMESTAMP,
   LASTSUCCESS = CURRENT TIMESTAMP,
   SYNCHPOINT = CURRENT TIMESTAMP
   WHERE SET_NAME = 'SET001' AND APPLY_QUAL = 'APPLY001'
   ```

7. Start the Apply program, or set ACTIVATE = 1 for the applicable replication subscription row in ASN.IBMSNAP_SUBS_SET as shown in the following SQL statements:

```
UPDATE ASN.IBMSNAP_SUBS_SET SET ACTIVATE = 1
WHERE SET_NAME = 'SET001' AND APPLY_QUAL = 'APPLY001'
```

Returning Control to Users with ASNDONE

ASNDONE is a user exit program that the Apply program invokes after set subscription processing completes, regardless of success or failure. You can modify ASNDONE to meet the requirements of your installation. For example, the user exit can examine the UOW table to discover rejected transactions and inititiate further actions, such as issuing a message or generating an alert.

To use ASNDONE:

1. Modify the ASNDONE file to meet your site's requirements.
2. Compile the program and place the executable in the appropriate directory.
3. Start the Apply program with the NONOTIFY invocation parameter to call the ASNLOAD program.

The parameters passed to ASNDONE are as follows:

- Control server name
- Set name
- Apply qualifier
- WHOS_ON_FIRST value
- Trace option
- Status value

Initiating a Forward Recovery with the Apply Program

In cases of an incomplete rollback or partially restored backup, you might need to copy a “window” of changes for completion.

To use the Apply program to limit the range of changes to those excluded from the rollback:

1. Determine how much data is lost.

   Check the IBMSNAP_LOGMARKER value in the target table. The highest value is the most recently committed change.

2. Use SQL to identify the range of changes. Either:

   - Use the WHERE clause on the Rows page of the Advanced Subscription Definition notebook.

     or

   - Manually update the PREDICATES column of the Common Subscription Targets Member table (ASN.IBMSNAP_SUBS_MEMBER).

   For example, if your timestamp shows the last committed change to be 941106174322, your SQL should reflect changes that occur after that time.

Troubleshooting

The following sections describe commonly known issues with Capture and Apply for MVS.

Problems Using the Capture Program

Before running Capture for MVS, ensure that you have all the PTFs for your level of DB2 and for the IFI installed on your DB2 subsystem. Specific Capture program problems are described below.

Problem

Capture for MVS does not start.

Ensure that APF authorization has been performed for all STEPLIB libraries as specified in the RUN JCL.

Problem

Capture for MVS is not capturing updates.

Any of the following could prevent the Capture program from capturing updates:

- Proper authorization was not granted to the administrator. Capture requires that the administrator have TRACE, MONITOR1, and MONITOR2 privileges.
- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting the Capture and Apply prorgrams was not used:
  1. Define replication sources and subscriptions before starting the Capture program.
  2. Start Capture for MVS and look for a message number ASN0100I (initialization completed) in the system console or in the ASN.IBMSNAP_TRACE table.
  3. Start the Apply program.

Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

Updates to the base table are not being replicated to the target table.

DB2 writes log information to the active log in multiples of 4KB VSAM control intervals (CI). In order for Capture to obtain these log records, there must be at least one CI written to the log. If your application has base table updates less than 4KB in a quiet DB2 (no other log updates), you do not see the update results in the change data table until the 4KB limit is reached. To see your updates you can either:

- Use -ARCHIVE LOG to flush out updates immediately.
- Use a filler table.

See Capture for MVS's program directory for information on creating a sample filler table. You can insert dummy rows into this table to reach the 4KB limit, so Capture can see other updates immediately.

Problem

I'm not sure if the Capture program is running successfully.

After you start the Capture and Apply programs, the Apply program performs a full refresh to populate the target tables. Then Capture writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which Capture starts to capture updates.

Updates captured from then on are placed in change data tables. They are eventually applied to target tables and pruned from the change data table. After Capture runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the progress made by Capture. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, the Apply program logs its information in the *userid*.IBMSNAP_APPLYTRAIL table.

Problem

Capture for MVS failed while using LE/370 environment.

Capture can run under both C/370 and LE/370 environments. If running under LE/370 environment, you must specify REGION=3M in the RUN step.

Problem

Capture for MVS issued message ASN0000E instead of the proper message number.

If you receive message ASN0000E, it implies that a generic message has been issued instead of a proper message. This happens if the specified VSAM message file in RUN JCL was not found. See the Capture for MVS program directory for information on installing the VSAM message file.

Problem

Capture for MVS terminated unexpectedly.

The Capture for MVS program terminates either because of a severe error, or when you issue the STOP command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0	STOP command issued
8	Error during initialization
12	Any other severe error

Problems Using the Apply Program

Before running the Apply program, ensure that:

- Replication sources and subscriptions have been defined.
- The Capture program has been started, and the ASN0100I initialization message has been issued (if you are running a Capture program).
- APF-authorization has been performed for all STEPLIB libraries of the Apply for MVSRUN JCL (if you are running Apply for MVS).
- Bind has been performed for all control, source, and target servers that the Apply program accesses.

When an error occurs while running the Apply program, the status field in IBMSNAP_SUBS_SET is set to -1 for that copy definition.

Errors as well as successful executions are logged in the IBMSNAP_APPLYTRAIL table. APPERRM contains the message text. Any SQL errors are also logged in this table. Error messages are also issued on the console.

Specific Apply program problems are described below.

Problem

I have performed a successful bind, but when running the Apply program, I still get SQLCODE -805, SQLSTATE 51002.

Make sure that the user ID has execute privilege on the Apply program packages, and make sure to bind the Capture package to the source server database and to bind both the Apply program packages to the control, source, and target server databases.

Problem

The DB2 log has filled to capacity because I copied a very large table.

If the error occured during a full refresh, you can use an alternative method to load large tables, described in “Loading Large Copies” on page 138.

If the error occured while applying changed data, then you can change the data blocking parameter to break down large blocks of changed data. See “Specifying Mini-Cycles for the Apply Program to Copy Committed Data” on page 115.

Problem

Capture was cold started, which caused the Apply program to perform a full refresh, but I don't want a full refresh.

If your target table is very large, and in cases where you have decided to use only your own load mechanism, you might want to suppress any future full refreshes of the Apply program. Set the DISABLE_REFRESH flag to 1 in ASN.IBMSNAP_REGISTER at the source server for the source table. In this case, the Apply program issues message ASN1016E.

Problem

A gap was detected, so the Apply program won't perform a full refresh of my target table.

Force a full refresh by resetting the LASTSUCCESS, SYNCHTIME, and SYNCHPOINT values in ASN.IBMSNAP_SUBS_SET to null.

I unsuccessfully tried to start a second Apply program instance.

You must run each instance with a unique Apply program qualifier.

Problem

I received error ASN1003 with SQLCODE = -1032 and SQLSTATE = 57019.

You must start the database manager before invoking the Apply program.

Problem

Apply components for DB2 Universal Database stops with an SQLCODE= -330, SQLSTATE=22517, "A string cannot be used, because its characters cannot be translated".

When copying between DB2 for MVS and DB2 Universal Database, the CCSID translation can cause an INSERT to fail if a translated value is longer than the DB2 column in which it will be inserted. Apply can generate an SQLCODE -330 when it tries to insert a translated COPY_TABLE column value from the refresh control table on a DB2 Universal Database copy server into the pruning control table on a DB2 for MVS source server.

For example, if you use the Korean character set with mixed data at both DB2 for MVS source server and DB2 Universal Database target server, the INSERT fails because the original string is in mixed data ASCII. When it is translated to EBCIDIC mixed data with the Korean character set, if the resulting string length is greater than 18 characters (the maximum COPY_TABLE length), the INSERT fails with an SQLCODE of -330.

Important: If you are running in a mixed environment, ensure you have installed the latest maintenance for the CCSID support of your DB2 for MVS program.

For more information on character translation, see the Character Conversion for Distributed Data chapter in *DB2 Version 4 Administration Guide, Volumes 1,2*.

Chapter 11. Capture and Apply for Windows NT and Windows 95

This chapter describes how to set up, operate, and troubleshoot Capture and Apply for Windows NT and Windows 95.

Read the following sections before reading the sections on operating the Capture program:

- "Setting Up the Capture and Apply Programs"
- "Specifying Tuning Parameters for the Capture Program" on page 148
- "Restrictions When Running the Capture Program" on page 149
- "Authorization for Running the Capture Program" on page 150

Setting Up the Capture and Apply Programs

Setting up consists of configuring the source, target, and control servers, and setting up NT services. The following sections provide instructions for configuring each server, providing end-user authentication at the source server, and setting up the NT Service Control Manager.

Configuring the Capture Program for Windows NT and Windows 95

1. Ensure that the user ID under which the Capture program is running has the required privileges:

   - Execute privilege on the Capture packages
   - DBADM or SYSADM privileges for the database

2. Log on with the user ID from the source server system.

3. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the source server database.

4. Prepare the source server database (source database) for roll-forward recovery by issuing the **update database configuration** command and the **backup database** command. For example:

   ```
   DB2 UPDATE DATABASE CONFIGURATION FOR database_alias USING LOGRETAIN ON
   DB2 BACKUP DATABASE database_alias
   ```

5. Change to the directory where the Capture program bind files are located, which is usually *drive*:\SQLLIB\BND.

6. Create and bind the Capture program package to the source server database by entering the following command:

   ```
   DB2 BIND @CAPTURE.LST ISOLATION UR BLOCKING ALL
   ```

   where *UR* specifies the list in uncommitted read format for greater performance.

   If your system does not support UR, substitute CS (cursor stability format) instead.

These commands create a list of packages, the names of which can be found in the CAPTURE.LST file.

Configuring the Apply Program for Windows NT and Windows 95

1. Ensure that the user ID under which the Apply program is running has the required privileges:

   - Execute privilege for the Apply packages
   - DBADM or SYSADM privileges for the database

2. Log on with the user ID from the target server system.

3. Change to the directory where the Apply program bind files are located, which is usually in *drive*:\SQLLIB\BND.

4. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the source server database.

5. Create and bind the Apply package to the source server database by entering both of the following commands:

   ```
   DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL
   ```

   ```
   DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL
   ```

   Where:

   CS The list in cursor stability format.

   UR The list in uncommitted read format.

   These commands create a list of packages, the names of which can be found in the APPLYCS.LST and APPLYUR.LST files.

6. Connect to the target server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the target server database.

7. Create and bind the Apply package to the target server database by entering both of the following commands:

   ```
   DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL GRANT PUBLIC
   ```

   ```
   DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL GRANT PUBLIC
   ```

   Note: Because the Apply program control tables use static SQL calls, the Apply bind process searches for nearly all of the control tables at each server that it is bound to, regardless of whether these control tables have any use or meaning at each server.

8. Repeat the connect and bind steps for each server that the Apply program connects to. You must bind the Apply program to the source, target and control servers.

Providing End-User Authentication at the Source Server

For end-user authentication to occur at the source server, you must provide a password file with the AUTH=SERVER scheme for the Apply program to use when connecting to the source server. It is advisable to limit read access of this file to the user ID that will run the Apply program.

Creating a password file:

The password file must meet the following criteria:

- Be named as shown:

  ```
  <APPLYQUAL><instname><CNTLSRVR>.PWD
  ```

  Where:

  APPLYQUAL
  The Apply qualifier upper case

  instname
  The instance name that the Apply program runs under. The value of DB2INSTANCE.

  CNTLSRVR
  The name of the control server in upper case

  For example: `DATADIRapply1REPRTDB.PWD`

  Note that this naming convention is the same as the .LOG file name and the .SPL file name, but with a file extension of PWD.

- Reside in the directory from which the Apply program starts
- Contain all server-name/password pairs for the file. This enables you to have a different (or the same) password at each server.The Apply user ID is used for all connections.
- Have one or more records using the following format:

  ```
  SERVER=server_name PWD=password
  ```

  The file cannot include blank lines or comment lines.

For more information about authentication and security, refer to the *IBM DB2 Universal Database Administration Guide*.

Setting Up the NT Service Control Manager

You can operate the Capture and Apply programs for Windows NT and Windows 95 by using the DB2 command processor, or by using the NT Service Control Manager (SCM). The SCM enables you to automatically start the Capture and Apply programs as services from the NT Control Panel.

To set up NT services:

1. Install the replication service by entering the fully-qualified path name of the ASNSERV executable as shown in the following example:

```
ASNINST D:\DIRECTORY\ASNSERV.EXE
```

2. Set up the service from the NT Control Panel.

   a. Click on the Services icon. The Services window appears.

   b. Select Replication and click on the STARTUP push button.

   c. Ensure that the startup type radio button indicates automatic.

   d. Specify the local user ID and password, then click on OK. The user ID must be the one that runs the Capture and Apply programs with the appropriate DB2 privileges.

3. Add the environment variable ASNPATH to specify the location of the Capture and Apply files.

   a. Select System from the NT Control Panel. The System Properties window appears.

   b. Click on the Environment tab.

   c. Type the ASNPATH string in the System Variable field as shown in the following example:

   ```
   ASNPATH=D:
   ```

   d. Click on OK.

4. Create an ASCII file to execute the Capture and Apply programs.

   a. Enter the following records in the file:

   db_name x:\\ASNCCP *parameters*

   db_name x:\\ASNAPPLY *parameters*

   where *db_name* is the name of your source database for the Capture program and control database for the Apply program, *x* is the drive, and *parameters* is one or more parameters as in the following example:

   ```
   DBNAME1 C:\CAPTURE\ASNCCP COLD TRACE
   DBNAME2 C:\APPLY\ASNAPPLY X 2
   ```

   b. Save the file to the following name:

   ```
   D:\NTSERV.ASN
   ```

After starting the service, the Capture and Apply programs run independently of ASNSERV. Consequently, stopping ASNSERV does not stop the Capture and Apply programs.

To remove ASNSERV from the NT Control Panel, use the ASNREMV program. After removing ASNSERV, Replication no longer appears on the Services window.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit

The number of minutes to keep the change data table rows and the unit-of-work (UOW) table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit

The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If the Capture program shuts itself down, you should perform a cold start if the database does not have or support an archive log.

Commit interval

The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if the Capture and Apply programs are running at the same time. This precaution helps to avoid locking overhead. If the Apply program is *not* running at the same time as the Capture program, you can set the commit interval no higher than the DB2 timeout interval.

Prune interval

The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is ignored if you start the Capture program with the NOPRUNE option.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

```
UPDATE TABLE ASN.IBMSNAP_CCPPARMS
SET RETENTION_LIMIT=number_of_minutes,
LAG_LIMIT=number_of_minutes,
COMMIT_INTERVAL=number_of_seconds,
PRUNE_INTERVAL=number_of_seconds
```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The Capture program does not capture any changes made by DB2 utilities.

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.

- Drop a replication source table.
- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have DBADM or SYSADM authority.

Operating Capture for Windows NT and Windows 95

The IBM Replication administrator user ID can use the commands in this section to operate Capture for Windows NT and Windows 95. Enter the commands or key combinations from an NT window.

This section explains how to perform the following Capture tasks:

- Starting
- Scheduling
- Stopping
- Suspending
- Resuming
- Reinitializing
- Pruning

Before You Start the Capture Program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where the Capture program is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS
  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC

  You can also create these control tables manually by running the DPCNTL.* file from the RUN SQL Files window.
- Bind the Capture program to the source servers from which the Capture program will capture changes. See “Configuring the Capture Program for Windows NT and Windows 95” on page 145 for more information.

- Enable the DATA CAPTURE CHANGES attribute for your source table using either the CREATE TABLE or ALTER TABLE statement. See Chapter 8, “Working with Replication Sources” on page 93 for more information.
- Decide whether you want to use the NT Service Control Manager (SCM) to automatically run the Capture program as an NT service. The following section provides information about operating the Capture program as an NT service.

Starting Capture for Windows NT and Windows 95

After you start the Capture program, it runs continuously until you stop it or it detects an error.

To start the Capture program using the NT services:

1. Select Replication from the NT Services window.
2. Click on the START push button. The Capture program starts according to the ASCII file information you provided.

You can also start the replication service by typing STRTSERV on the NT command line.

To start the Capture program using the DB2 command window:

1. If you created one or more DB2 for NT instances, from the System window, set the DB2INSTANCE environment variable to the DB2 for NT instance with which you plan to run the Capture program:

   ```
   SET DB2INSTANCE=database_instance_name
   ```

   While the Capture program is running, a file with the name <*database_instance_name*><*database_name*>.CCP is created in the directory from which the Capture program is invoked. This file is a log file for the messages issued by the Capture program; these messages are also recorded in the ASN.IBMSNAP_TRACE table.

2. Enter the ASNCCP command from the Capture for Windows NT or Windows 95 window where you issued the SET command. The syntax is:

```
►►──ASNCCP──┬────────────┬──┬────────┬──┬─────────┬──┬─────────┬──────────►◄
            └─Src_server─┘  ├─WARM───┤  ├─PRUNE───┤  ├─NOTRACE─┤
                            ├─WARMNS─┤  └─NOPRUNE─┘  └─TRACE───┘
                            └─COLD───┘
```

Table 11 on page 152 defines the optional parameters.

Table 11. ASNCCP Command Parameter Definitions for Windows NT or Windows 95

Parameter	Definition
Src_server	Source server name positional parameter that must be the first parameter if entered. If not specified, the value from the DB2DBDFT environment variable will be used.
WARM (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. See “Warm and Cold Starts” on page 156 for more information.
WARMNS	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With WARMNS, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, the Capture program terminates instead of switching to a cold start as when WARM is specified. See “Warm and Cold Starts” on page 156 for more information.
COLD	The Capture program starts by deleting all rows in its change data, UOW, pruning control, and trace tables during initialization. See “Warm and Cold Starts” on page 156 for more information.
PRUNE (default)	The Capture program automatically prunes the change data and UOW tables.
NOPRUNE	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the PRUNE command. See the PRUNE command for more information.
NOTRACE (default)	No trace information is written.
TRACE	Writes debug trace messages to the standard output, *stdout*.

Scheduling Capture for Windows NT and Windows 95

Use the AT command to start the Capture program at a specific time. For example, the following command string starts the Capture program for Windows NT at 15:00.

```
c:\>AT 15:00 /interactive "c:\SQLLIB\BIN\db2cmd.exe c:\CAPTURE\asnccp.exe cold"
```

Note: The Windows NT or Windows 95 Schedule Service must be started before using the AT command.

Stopping Capture for Windows NT and Windows 95

If you started the Capture program as an NT service, stop ASNSERV by selecting Replication from the NT Services window and clicking the STOP push button. After the stop message appears, the status field becomes blank.

If you started the Capture program from the DB2 command window, enter the following command:

```
►►──ASNCMD──STOP──────────────────────────────────────────────────►◄
```

or one of the following:

- Ctrl+C

- Ctrl+Break

Purpose

Use the STOP command or the key combination to stop the Capture program in an orderly way and commit the log records that it processed up to that point.

Stop the Capture program before:

- Removing an existing replication source
- Opening an existing replication source and modifying the **Enable Refresh and Update** field

Usage

- To use the command, do the following from a different window than where the Capture program is running:
  1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
  2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
  3. Enter the command.
- If you want to use one of the key combinations, press it from the same NT window in which the Capture program is running.

Suspending Capture for Windows NT and Windows 95

```
►►──ASNCMD──SUSPEND──────────────────────────────────────────────►◄
```

Purpose

Use the SUSPEND command to relinquish Windows NT or Windows 95 resources to operational transactions during peak periods without damaging the Capture program environment.

This command suspends the Capture program until you issue the RESUME command.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Attention: Do not use SUSPEND when canceling a replication source. Instead, stop the Capture program using Ctrl+C.

Resuming Capture for Windows NT and Windows 95

```
►►──ASNCMD──RESUME─────────────────────────────────────────────────────►◄
```

Purpose

Use the RESUME command to restart the Capture program if you suspended it using the SUSPEND command.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Reinitializing Capture for Windows NT and Windows 95

```
►►──ASNCMD──REINIT─────────────────────────────────────────────────────►◄
```

Purpose

Use the REINIT command to make the Capture program reread the ASN.IBMSNAP_REGISTER tables or the ASN.IBMSNAP_CCPPARMS table while it is running. Use this command when you add a new replication source with the Control Center while the Capture program is running.

This command ensures that the Capture program recognizes the new replication source. The Capture program also rereads the ASN.IBMSNAP_CCPPARMS table for any changes made to the tuning parameters.

Attention: Do not use REINIT to reinitialize the Capture program after canceling a replication source or dropping a replication source table while the Capture program is running. Instead, stop the Capture program and restart it using the WARM option.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.

2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Pruning the Change Data and Unit-of-Work Tables

```
►►──ASNCMD──PRUNE──────────────────────────────────────────────────►◄
```

Purpose

Use the PRUNE command to prune the CD table and UOW table if you used the NOPRUNE invocation parameter to disable pruning when you started the Capture program. This command prunes the CD and UOW tables once. The Capture program issues the message ASN0124I when the command is successfully queued.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

During pruning, if you stop or suspend the Capture program, pruning does not resume after you enter the RESUME command. You must enter the PRUNE command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──ASNCMD──GETLSEQ────────────────────────────────────────────────►◄
```

Purpose

Use the GETLSEQ command to provide the current log sequence number and the timestamp. You can use this number to determine how far the Capture program has read the DB2 log.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.

2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Warm and Cold Starts

This section explains how the Capture program handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the warm start table, ASN.IBMSNAP_WARM_START, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, UOW table, or change data table.

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, an operator might cancel the Capture program or stop DB2. In this case, the Capture program uses the change data, UOW, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (The Apply program performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued. For information about handling gap messages, see “Problems Using the Apply Program” on page 163 and read the portion about forcing the Apply program to perform a full refresh.)

Automatic Cold Starts

Sometimes the Capture program automatically switches to a cold start, even if you specified a warm start (not WARMNS). For example:

- The warm start log sequence lags behind the current log sequence by more than the LAG_LIMIT value (as specified in the tuning parameters table ASN.IBMSNAP_CCPPARMS).
- You invoke the Capture program for the first time.

  The first time that you invoke the Capture program, you see message ASN0102W, indicating that the warm start failed. the Capture program switches to a cold start. You can ignore this message when first invoking the Capture program.

For each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, or if someone brings down the DB2 table space containing the change data table. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Operating Apply for Windows NT and Windows 95

The IBM Replication administrator can use the commands in the following sections to perform the following Apply for Windows NT tasks:

- Starting
- Scheduling
- Stopping

Before You Start the Apply Program

Before you start the Apply program, ensure that:

- The administrator has created the control tables.
- A password file has been created, if necessary, for end-user authentication at the source server. See “Providing End-User Authentication at the Source Server” on page 147 for more information.
- At least one subscription is created and activated.
- The Apply for Windows NT package is created. See “Configuring the Apply Program for Windows NT and Windows 95” on page 146 for information on bind programs to create the Apply packages.
- You have the proper authorization. See “Authorization Requirements for the Capture Program” on page 66 for information about authorization for the Apply program.

Starting Apply for Windows NT and Windows 95

After you start the Apply program, it runs continuously until:

- You stop it in an orderly way.
- An unexpected error or failure occurs.

To start the Apply program using the NT services:

1. Select Replication from the NT Services window.
2. Click on the START push button. The Apply program starts according to the ASCII file information you provided.

You can also start the replication service by typing STRTSERV on the NT command line.

To start the Apply program using the DB2 command window:

1. Ensure that you set the DB2 instance as shown:

   ```
   SET DB2INSTANCE=db2_instance_name
   ```

2. Enter the ASNAPPLY command and one or more options, if desired, from the Windows NT or Windows 95 window.

```
►►──ASNAPPLY──Apl_qual──┬───────────┬──┬───────────┬──┬──────────┬──────────►
                        └─Ctrl_serv─┘  ├─LOADXit───┤  ├─INAMsg───┤
                                       └─NOLOADXit─┘  └─NOINAMsg─┘

►──┬─────────┬──┬──────────┬───────────────────────────────────────────────►◄
   ├─NOTRC───┤  ├─NOTIFY───┤
   ├─TRCERR──┤  └─NONOTIFY─┘
   └─TRCFLOW─┘
```

Table 12 defines the optional positional parameters.

Table 12. ASNAPPLY Parameter Definitions for NT

Parameter	Definition
Apl_qual	Specifies the Apply qualifier that the Apply instance uses to identify the subscriptions to be served. This positional parameter is required as the first parameter.
Ctrl_serv	Specifies the name of the server where the replication control tables will reside. If you do not specify this optional positional parameter, the default is the default database or the value of DB2DBDFT for DB2 for Windows NT or Windows 95.
LOADXit	Specifies that the Apply program is to invoke ASNLOAD, an exit program that can call an IBM or multivendor utility, to initialize a target table. This keyword parameter is optional.
NOLOADXit (default)	Specifies that the Apply program will not invoke ASNLOAD.
INAMsg (default)	Specifies that the Apply program is to issue a message when it is inactive.
NOINAMsg	Specifies that the Apply program will not issue this message. This keyword parameter is optional.
NOTRC (default)	Specifies that the Apply program does not generate a trace.
TRCERR	Specifies that the Apply program generates a trace that contains only error information. This keyword parameter is optional.
TRCFLOW	Specifies that the Apply program generates a trace that contains error and execution flow information. This keyword parameter is optional.
NOTIFY	Specifies that the Apply program is to invoke ASNDONE, an exit program that returns control to the user when the Apply program processing ends. This keyword parameter is optional.
NONOTIFY (default)	Specifies that the Apply program will not invoke ASNDONE.

When Apply for Windows NT or Windows 95 executes, it generates two files:

- *<userid><database_instance_name><database_name>*.LOG

This file is created in the directory where Apply for Windows NT or Windows 95 is invoked. It contains messages issued by Apply for Windows NT or Windows 95.

- *<userid><database_instance_name><database_name>*.SPL

  This file is created in the directory where Apply for Windows NT or Windows 95 is invoked. It is an internal file used by Apply for Windows NT or Windows 95. Do not erase it.

Scheduling Apply for Windows NT and Windows 95

Use the Windows NT or Windows 95 AT command to start the Apply program at a specific time. For example, the following command string starts Apply for Windows NT at 15:00.

```
c:\>AT 15:00 /interactive "c:\SQLLIB\BIN\db2cmd.exe
    c:\SQLLIB\BIN\asnapply.exe qualid1 cntldb"
```

Note: Before entering the AT command, the Schedule service should already be started.

Stopping Apply for Windows NT and Windows 95

Use the following command to stop Apply for Windows NT or Windows 95.

```
►►──ASNASTOP──Apply_qualifier──────────────────────────────────►◄
```

You can alternatively use one of the following key combinations from the window where the Apply program is running:

- Ctrl+C
- Ctrl+Break

Purpose

Use the ASNASTOP command or the key combination to stop the Apply program in an orderly way and commit the log records that it processed up to that point.

Usage

- To use the command, do the following from a different window than where the Apply program is running:
  1. Set environment variable DB2INSTANCE to the value that was set when the Apply program was started.
  2. Set environment variable DB2DBDFT to the source server specified when the Apply program was started (or the DB2DBDFT value that was used when the Apply program was started).
  3. Enter the command.
- To use one of the key combinations, press it from the same NT window in which the Apply program is running.

Additional Apply Program Operations

The following section provides information about performing the additional Apply program functions of scheduling subscriptions, refreshing target tables, returning control to users, and initiating a forward recovery.

Scheduling Subscriptions with the Event Table

You can include an event name in your replication subscription. Your subscription cycles can run according to a timer, an event occurrence, or both. See “Setting the Copying Schedule: Time or Event Based” on page 113 for more information about the purpose and usage of event scheduling.

You can schedule a subscription by manipulating the event table. The DDL statement for the event table appears as follows:

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_EVENT (
  EVENT_NAME CHAR(18) NOT NULL,
  EVENT_TIME TIMESTAMP NOT NULL,
  END_OF_PERIOD TIMESTAMP);
```

For example, to start the Apply program at 6:00 p.m.. on 9/24/97 and copy changes from the source server up to 4:00 p.m.. on the same date, you would insert the following values in ASN.IBMSNAP_SUBS_EVENT:

```
('EVENTA','1997-09-24-18.00.00.00000','1997-09-24-16.00.00.00000'),
```

A subscription becomes eligible for processing when EVENT_TIME falls between LASTSUCCESS and the control server's CURRENT TIMESTAMP. The END_OF_PERIOD column enables you to specify a timestamp upper limit for change data.

Refreshing Target Tables with ASNLOAD

ASNLOAD is a user exit program that the Apply program invokes when you specify the LOADX invocation parameter. The Apply program calls the ASNLOAD program whenever it performs a full refresh of a target table.

You can use the ASNLOAD program as shipped with the Apply program, or you can modify it. As shipped, ASNLOAD uses the EXPORT utility to export data from the source table and uses the LOAD utility to fully refresh the target table.

You can modify ASNLOAD to call any IBM or vendor utility. See the prolog section in the sample program (ASNLOAD.SMP) for instructions on how to modify the program to meet the requirements of your installation.

You must use ASNLOAD to fully refresh tables with referential integrity constraints in order to bypass referential integrity checking.

Files Generated

If you invoke ASNLOAD, the following files are generated:

- ASNA<*userid*><*database_instance_name*><*cntl_server*>.IXF

  This file contains the data exported from the source.

- ASNAEXPT<*userid*><*database_instance_name*><*cntl_server*>.MSG

  This file contains error, warning, or informational messages issued by the EXPORT APIs.

- ASNAIMPT<*userid*><*database_instance_name*><*cntl_server*>.MSG

  This file contains error, warning, or informational messages issued by the LOAD APIs.

Error Handling

If an error occurs while the Apply program calls ASNLOAD, or if ASNLOAD returns a nonzero return code, the Apply program issues a message, stops processing that subscription, and processes the next subscription.

Returning Control to Users with ASNDONE

If you specify the NOTIFY parameter when starting the Apply program, the user exit program ASNDONE is called after subscription processing completes, regardless of success or failure. You can modify ASNDONE to meet the requirements of your installation. For example, the user exit can examine the UOW table to discover rejected transactions and initiate further actions, such as issuing a message or generating an alert.

See the prolog section in the sample program (ASNDONE.SMP) for instructions on how to modify the program.

Initiating a Forward Recovery with the Apply Program

In cases of an incomplete rollback or partially restored backup, you might need to copy a “window” of changes for completion.

To use the Apply program to limit the range of changes to those excluded from the rollback:

1. Determine how much data is lost.

   Check the IBMSNAP_LOGMARKER value in the point-in-time table. The highest value is the most recently committed change.

2. Use SQL to identify the range of changes. Either:

   - Use the WHERE clause on the Rows page of the Advanced Subscription Definition notebook.

     or

- Manually update the PREDICATES column of the Common Subscription Targets Member table (ASN.IBMSNAP_SUBS_MEMBER).

For example, if your timestamp shows the last committed change to be 941106174322, your SQL should reflect changes that occur after that time.

Troubleshooting

The following sections describe commonly known issues with Capture and Apply for Windows NT and Windows 95.

Problems Using the Capture Program

Problem

Capture for Windows NT is not capturing updates.

Any of the following could prevent the Capture program from capturing updates:

- Proper authorization was not granted to the user ID running the Capture program. The Capture program requires DBADM or SYSADM privileges.
- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting the Capture and Apply programs was not used:
  1. Define replication sources and subscriptions before starting the Capture program.
  2. Start the Capture program and look for message number ASN0100I (initialization completed) in the system console or in the ASN.IBMSNAP_TRACE table.
  3. Start the Apply program.

  Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

I'm not sure if the Capture program is running successfully.

After you start the Capture and Apply programs, the Apply program performs a full refresh to populate the target tables. Then the Capture program writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which the Capture program starts to capture updates.

Updates captured from then on are placed in change data tables. They are eventually applied to target tables and pruned from the change data table. After the Capture program runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the progress made by the Capture program. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, the Apply program logs its information in the *userid*.IBMSNAP_APPLYTRAIL table.

Problem

Capture for Windows NT terminated unexpectedly.

The Capture for Windows NT program terminates either because of a severe error, or when you issue the STOP command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0 STOP command issued
8 Error during initialization
12 Any other severe error

Problems Using the Apply Program

Before running the Apply program, ensure that:

- Replication sources and subscriptions have been defined.
- The Capture program has been started, and the ASN0100I initialization message has been issued (if you are running a Capture program).
- Bind has been performed for all control, source, and target servers that the Apply program accesses.

When an error occurs while running the Apply program, the status field in IBMSNAP_SUBS_SET is set to -1 for that copy definition.

Errors as well as successful executions are logged in the IBMSNAP_APPLYTRAIL table. APPERRM contains the message text. Any SQL errors are also logged in this table. Error messages are also issued on the console.

Specific Apply program problems are described below.

Problem

I have performed a successful bind, but when running the Apply program, I still get SQLCODE -805, SQLSTATE 51002.

Make sure that the user ID has execute privilege on the Apply packages, and make sure to bind the Capture package to the source server database and to bind both Apply packages to the control, source, and target server databases.

Problem

The DB2 log has filled to capacity because I copied a very large table.

If the error occurred during a full refresh, you can use alternative methods to load large tables. You can either use the ASNLOAD exit, described in Table 12 on page 158, or you can perform your own load, described in “Loading Large Copies” on page 138.

If the error occurred while applying changed data, you can change the data blocking parameter to break down large blocks of changed data. See “Specifying Mini-Cycles for the Apply Program to Copy Committed Data” on page 115.

Problem

The Capture program was cold started, which caused the Apply program to perform a full refresh, but I don't want a full refresh.

If your target table is very large, and in cases where you have decided to use only your own load mechanism, you might want to suppress any future full refreshes of the Apply program. Set the DISABLE_REFRESH flag to 1 in ASN.IBMSNAP_REGISTER at the source server for the source table. In this case, the Apply program issues message ASN1016E.

Problem

A gap was detected, so the Apply program won't perform a full refresh of my target table.

Force a full refresh by resetting the LAST_SUCCESS, SYNCHTIME, and SYNCHPOINT values in ASN.IBMSNAP_SUBS_SET to null.

I unsuccessfully tried to start a second Apply program instance.

You must run each instance with a unique Apply qualifier.

Problem

I received error ASN1003 with SQLCODE = -1032 and SQLSTATE = 57019.

You must start the database manager before invoking the Apply program.

Problem

The Apply components for DB2 Universal Database stop with an SQLCODE= -330, SQLSTATE=22517, "A string cannot be used, because its characters cannot be translated".

When copying between DB2 for MVS and DB2 Universal Database, the CCSID translation can cause an INSERT to fail if a translated value is longer than the DB2 column in which it will be inserted. The Apply program can generate an SQLCODE -330 when it tries to insert a translated COPY_TABLE column value from the refresh control table on a DB2 Universal Database target server into the pruning control table on a DB2 for MVS source server.

For example, if you use the Korean character set with mixed data at both DB2 for MVS source server and DB2 Universal Database target server, the INSERT fails because the original string is in mixed data ASCII. When it is translated to EBCIDIC mixed data with the Korean character set, if the resulting string length is greater than 18 characters (the maximum COPY_TABLE length), the INSERT fails with an SQLCODE of -330.

Important: If you are running in a mixed environment, ensure you have installed the latest maintenance for the CCSID support of your DB2 for MVS program.

For more information on character translation, see the Character Conversion for Distributed Data chapter in *DB2 Version 4 Administration Guide, Volumes 1,2*.

Chapter 12. Capture and Apply for OS/2

This chapter describes how to set up, operate, and troubleshoot Capture and Apply for OS/2.

Read the following sections before reading the sections on operating the Capture and Apply programs.

- "Setting Up the Capture and Apply Programs"
- "Specifying Tuning Parameters for the Capture Program" on page 168
- "Restrictions When Running the Capture Program" on page 169
- "Authorization for Running the Capture Program" on page 170

Setting Up the Capture and Apply Programs

Setting up consists of configuring the source, target, and control servers. The following sections provide instructions for configuring each server.

Configuring the Capture Program for OS/2

1. Ensure that the user ID under which the Capture program is running has the required privileges:

   - Execute privilege on the Capture packages
   - DBADM or SYSADM privileges for the source, control, and target servers

2. Log on with the user ID from the source server system.

3. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the source server database.

4. Prepare the source server database (source database) for roll-forward recovery by issuing the **update database configuration** command and the **backup database** command. For example:

   ```
   DB2 UPDATE DATABASE CONFIGURATION FOR database_alias USING LOGRETAIN ON
   DB2 BACKUP DATABASE database_alias
   ```

5. Change to the directory where the Capture program bind files are located, which is usually *drive*:\SQLLIB\BND.

6. Create and bind the Capture program package to the source server database by entering the following command:

   ```
   DB2 BIND @CAPTURE.LST ISOLATION UR BLOCKING ALL
   ```

   where *UR* specifies the list in uncommitted read format for greater performance.

   If your system does not support UR, substitute CS (cursor stability format) instead.

   These commands create a list of packages, the names of which can be found in the CAPTURE.LST file.

Configuring the Apply Program for OS/2

1. Ensure that the user ID under which Apply is running has the required privileges:
   - Execute privilege for Apply packages
   - DBADM or SYSADM privileges for the database
2. Log on with the user ID from the target server system.
3. Change to the directory where the Apply program bind files are located, which is usually in *drive*:\SQLLIB\BND.
4. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the source server database.
5. Create and bind the Apply package to the source server database by entering both of the following commands:

   ```
   DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL
   ```

   ```
   DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL
   ```

   Where:

   CS The list in cursor stability format.

   UR The list in uncommitted read format.

   These commands create a list of packages, the names of which can be found in the APPLYCS.LST and APPLYUR.LST files.
6. Connect to the target server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the target server database.
7. Create and bind the Apply package to the target server database by entering both of the following commands:

   ```
   DB2 BIND @APPLYCS.LST ISOLATION CS BLOCKING ALL GRANT PUBLIC
   ```

   ```
   DB2 BIND @APPLYUR.LST ISOLATION UR BLOCKING ALL GRANT PUBLIC
   ```

   Note: Because the Apply control tables use static SQL calls, the Apply bind process searches for nearly all of the control tables at each server that it is bound to, regardless of whether these control tables have any use or meaning at each server.
8. Repeat the connect and bind steps for each server that the Apply program connects to. You must bind the Apply program to the source, target and control servers.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit
The number of minutes to keep the change data table rows and the UOW table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit
The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If the Capture program shuts itself down, you should perform a cold start if the database does not have or support an archive log.

Commit interval
The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if the Capture and Apply programs are running at the same time. This precaution helps to avoid locking overhead. If the Apply program is *not* running at the same time as the Capture program, you can set the commit interval no higher than the DB2 timeout interval.

Prune interval
The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is ignored if you start the Capture program with the NOPRUNE option.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

```
UPDATE TABLE ASN.IBMSNAP_CCPPARMS
SET RETENTION_LIMIT=number_of_minutes,
LAG_LIMIT=number_of_minutes,
COMMIT_INTERVAL=number_of_seconds,
PRUNE_INTERVAL=number_of_seconds
```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.
- Drop a replication source table.

- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Other Capture program restrictions are:

- The Capture program does not capture any changes made by DB2 utilities.
- Capture for OS/2 does not support large object and LONG data types.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have DBADM or SYSADM authority.

Operating Capture for OS/2

The IBM Replication administrator user ID can use the commands in this section to operate Capture for OS/2. Enter the commands or key combinations from an OS/2 window.

This section explains how to perform the following Capture for OS/2 tasks:

- Starting
- Scheduling
- Stopping
- Suspending
- Resuming
- Reinitializing
- Pruning

Before You Start the Capture Program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where the Capture program is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS
  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC

  You can also create these control tables manually by running the DPCNTL.* file from the RUN SQL Files window.

- Bind the Capture program to the source servers from which the Capture program will capture changes. “Configuring the Capture Program for OS/2” on page 167 for more information.
- Enable the DATA CAPTURE CHANGES attribute for your source table using either the CREATE TABLE or ALTER TABLE statement.

Starting Capture for OS/2

After you start the Capture program, it runs continuously until you stop it or it detects an error.

To start the Capture program:

1. If you created one or more DB2 for OS/2 instances, use the SET command to set the DB2INSTANCE environment variable to the DB2 for OS/2 instance with which you plan to run the Capture program:

   ```
   SET DB2INSTANCE=database_instance_name
   ```

   While the Capture program is running, a file with the name *database_name*.CCP is created in the directory from which the Capture program is invoked. This file is a log file for the messages issued by the Capture program; these messages are also recorded in the ASN.IBMSNAP_TRACE table.

2. Enter the ASNCCP command from the OS/2 window where you issued the SET command. The syntax is:

```
►►──ASNCCP──┬────────────┬──┬────────┬──┬─────────┬──┬─────────┬──────►◄
            └─Src_server─┘  ├─WARM───┤  ├─PRUNE───┤  ├─NOTRACE─┤
                            ├─WARMNS─┤  └─NOPRUNE─┘  └─TRACE───┘
                            └─COLD───┘
```

Table 13 on page 172 defines the optional parameters.

Table 13. ASNCCP Command Parameter Definitions for OS/2

Parameter	Definition
Src_server	Source server name positional parameter that must be the first parameter if entered. If not specified, the value from the DB2DBDFT environment variable will be used.
WARM (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. See “Warm and Cold Starts” on page 175 for more information.
WARMNS	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With WARMNS, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, the Capture program terminates instead of switching to a cold start as when WARM is specified. See “Warm and Cold Starts” on page 175 for more information.
COLD	The Capture program starts by deleting all rows in its change data table, UOW table, pruning control table, and trace table during initialization. Refer to “Warm and Cold Starts” on page 175 for more information.
PRUNE (default)	The Capture program automatically prunes the change data and UOW tables.
NOPRUNE	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the PRUNE command. See the PRUNE command for more information.
NOTRACE (default)	No trace information is written.
TRACE	Writes debug trace messages to the standard output, *stdout*.

Scheduling Capture for OS/2

Use the Alarms program in the OS/2 Productivity set to start Capture for OS/2 at a specific time.

Stopping Capture for OS/2

```
►►──ASNCMD──STOP──────────────────────────────────────────────►◄
```

or one of the following:

- Ctrl+C
- Ctrl+Break

Purpose

Use the STOP command or the key combination to stop the Capture program in an orderly way and commit the log records that it processed up to that point.

Stop the Capture program before:

- Removing an existing replication source
- Opening an existing replication source and modifying the **Enable Refresh and Update** field

Usage

- To use the command, do the following from a different OS/2 window than where the Capture program is running:
  1. Set the environment variable DB2INSTANCE to the value that was set when the Capture program was started.
  2. Set the environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
  3. Enter the command.
- If you want to use one of the key combinations, press it from the same OS/2 window in which the Capture program is running.

Suspending Capture for OS/2

```
►►──ASNCMD──SUSPEND──────────────────────────────────────────────────►◄
```

Purpose

Use the SUSPEND command to relinquish OS/2 resources to operational transactions during peak periods without damaging the Capture program environment.

This command suspends the Capture program until you issue the RESUME command.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Attention: Do not use SUSPEND when canceling a replication source. Instead, stop the Capture program using Ctrl+C.

Resuming Capture for OS/2

```
►►──ASNCMD──RESUME───────────────────────────────────────────────────►◄
```

Purpose

Use the RESUME command to restart the Capture program if you suspended it using the SUSPEND command.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Reinitializing Capture for OS/2

```
►►──ASNCMD──REINIT───────────────────────────────────────────────────►◄
```

Purpose

Use the REINIT command to make the Capture program reread the ASN.IBMSNAP_REGISTER table or the ASN.IBMSNAP_CCPPARMS table while it is running. Use this command when you add a new replication source with Control Center while the Capture program is running.

This command ensures that the Capture program recognizes the new replication source. The Capture program also rereads the ASN.IBMSNAP_CCPPARMS table for any changes made to the tuning parameters.

Attention:

Do not use REINIT to reinitialize the Capture program after canceling a replication source or dropping a replication source table while the Capture program is running. Instead, stop the Capture program and restart it using the WARM option.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Pruning the Change Data and Unit-of-Work Tables

```
►►──ASNCMD──PRUNE────────────────────────────────────────────────────►◄
```

Purpose

Use the PRUNE command to prune the CD and UOW tables if you used the NOPRUNE invocation parameter to disable pruning when you started the Capture program. This command prunes the CD and UOW tables once. The Capture program issues the message ASN0124I when the command is successfully queued.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

During pruning, if you stop or suspend the Capture program, pruning does not resume after you enter the RESUME command. You must enter the PRUNE command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──ASNCMD──GETLSEQ──────────────────────────────────────────────────►◄
```

Purpose

Use the GETLSEQ command to provide the current log sequence number and the timestamp. You can use this number to determine how far the Capture program has read the DB2 log.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Warm and Cold Starts

This section explains how the Capture program handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the ASN.IBMSNAP_WARM_START table, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, UOW table, or change data tables.

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, an operator might cancel the Capture program or stop DB2. In this case, the Capture program uses the change data, UOW, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (The Apply program performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued. For information about handling gap messages, see “Problems Using the Apply Program” on page 182 and read the portion about forcing the Apply program to perform a full refresh.)

Automatic Cold Starts

Sometimes the Capture program automatically switches to a cold start, even if you specified a warm start (not WARMNS). For example:

- The warm start log sequence lags behind the current log sequence by more than the LAG_LIMIT value (as specified in the tuning parameters table ASN.IBMSNAP_CCPPARMS).
- You invoke the Capture program for the first time.

  The first time that you invoke the Capture program, you see message ASN0102W, indicating that the warm start failed. The Capture program switches to a cold start. You can ignore this message when first invoking the Capture program.

For each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, or if someone brings down the DB2 table space containing the change data table. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Operating Apply for OS/2

The IBM Replication administrator can use the commands in the following sections to perform the following Apply for OS/2 tasks:

- Starting
- Scheduling
- Stopping

Before You Start theApply Program

Before you start the Apply program, ensure that:

- The administrator has created the control tables.
- At least one subscription is created and activated.
- The Apply for OS/2 package is created. See “Configuring the Apply Program for OS/2” on page 168 for information on bind programs to create the Apply for OS/2 package.
- You have the proper authorization. See “Authorization Requirements for the Capture Program” on page 66 for information about authorization for the Apply program.

Starting Apply for OS/2

After you start Apply, it runs continuously until:

- You stop it in an orderly way.
- An unexpected error or failure occurs.

To start Apply:

Perform the following steps from an OS/2 window:

1. Log on with the IBM Replication user ID.
2. Ensure that you set the DB2 instance name as shown:

   ```
   SET DB2INSTANCE=db2_instance_name
   ```

3. Enter the ASNAPPLY command and one or more options, if desired, from the OS/2 window.

```
►►──ASNAPPLY──Apl_qual──┬───────────┬──┬────────────┬──┬───────────┬──────►
                        └─Ctrl_serv─┘  ├─LOADXit────┤  ├─INAMsg────┤
                                       └─NOLOADXit──┘  └─NOINAMsg──┘

►──┬──────────┬──┬──────────┬──────────────────────────────────────────►◄
   ├─NOTRC────┤  ├─NOTIFY───┤
   ├─TRCERR───┤  └─NONOTIFY─┘
   └─TRCFLOW──┘
```

Table 14 on page 178 defines the optional positional parameters.

Table 14. ASNAPPLY Parameter Definitions for OS/2

Parameter	Definition
Apl_qual	Specifies the Apply qualifier that the Apply instance uses to identify the subscriptions to be served. This positional parameter is required as the first parameter.
Ctrl_serv	Specifies the name of the control server where the subscriptions to be served by this instance of the Apply program reside. If you specify this optional positional parameter, it must be the second parameter. If you do not specify this parameter, the default is the default database or the value of DB2DBDFT for DB2 for OS/2.
LOADXit	Specifies that the Apply program is to invoke ASNLOAD, a program that can call an IBM or multivendor utility, to initialize a target table. This keyword parameter is optional.
NOLOADXit (default)	Specifies that the Apply program will not invoke ASNLOAD.
INAMsg (default)	Specifies that the Apply program is to issue a message when the Apply program is inactive.
NOINAMsg	Specifies that the Apply program does not issue a message when the Apply program is inactive. This is an optional keyword parameter.
NOTRC (default)	Specifies that the Apply program does not generate a trace.
TRCERR	Specifies that the Apply program generates a trace that contains only error information. This is an optional keyword parameter.
TRCFLOW	Specifies that the Apply program generates a trace that contains error and execution flow information. This is an optional keyword parameter.
NOTIFY	Specifies that the Apply program is to invoke ASNDONE, an exit program that returns control to the user when Apply program processing ends. This is an optional keyword parameter.
NONOTIFY (default)	Specifies that the Apply program does not invoke ASNDONE.

When Apply for OS/2 executes, it generates two files:

- *userid*.LOG

  This file is created in the directory where Apply for OS/2 is invoked. It contains messages issued by Apply for OS/2.

- *userid*.SPL

  This file is created in the directory where Apply for OS/2 is invoked. It is an internal file used by Apply for OS/2. Do not erase it.

Scheduling Apply for OS/2

Use the Alarms program in the OS/2 Productivity set.

Stopping Apply for OS/2

Use the following command to stop Apply for OS/2.

```
►►──ASNASTOP──Apply_qualifier──────────────────────────────────────►◄
```

You can alternatively use one of the following key combinations from the window where the Apply program is running:

- Ctrl+C
- Ctrl+Break

Purpose

Use the ASNASTOP command or the key combination to stop the Apply program in an orderly way and commit the log records that it processed up to that point.

Usage

To use the command, do the following from a different window than where the Apply program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Apply program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Apply program was started (or the DB2DBDFT value that was used when the Apply program was started).
3. Enter the command.

   If you want to use one of the key combinations, press it from the same NT window in which the Apply program is running.

Additional Apply for OS/2 Operations

The following section provides information about performing the additional Apply program functions of scheduling subscriptions, refreshing target tables, returning control to users, and initiating a forward recovery.

Scheduling Subscriptions with the Event Table

You can include an event name in your replication subscription. Your subscription cycles can run according to a timer, an event occurrence, or both. See “Setting the Copying Schedule: Time or Event Based” on page 113 for more information about the purpose and usage of event scheduling.

You can schedule a subscription by manipulating the event table. The DDL statement for the event table appears as follows:

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_EVENT (
  EVENT_NAME CHAR(18) NOT NULL,
  EVENT_TIME TIMESTAMP NOT NULL,
  END_OF_PERIOD TIMESTAMP);
```

For example, to start the Apply program at 6:00 p.m. on 9/24/97 and copy changes from the source server up to 4:00 p.m. on the same date, you would insert the following values in ASN.IBMSNAP_SUBS_EVENT:

```
('EVENTA','1997-09-24-18.00.00.00000','1997-09-24-16.00.00.00000'),
```

A subscription becomes eligible for processing when EVENT_TIME falls between LASTSUCCESS and the control server's CURRENT TIMESTAMP. The END_OF_PERIOD column enables you to specify a timestamp upper limit for change data.

Refreshing Target Tables with ASNLOAD

ASNLOAD is a user exit program that the Apply program invokes when you specify the LOADX invocation parameter. The Apply program calls the ASNLOAD program whenever it performs a full refresh of a target table.

You can use the ASNLOAD program as shipped with the Apply program, or you can modify it. As shipped, ASNLOAD uses the EXPORT utility to export data from the source table and uses the LOAD utility to fully refresh the target table.

You can modify ASNLOAD to call any IBM or vendor utility. See the prolog section in the sample program (ASNLOAD.SMP) for instructions on how to modify the program to meet the requirements of your installation.

You must use ASNLOAD to fully refresh tables with referential integrity constraints in order to bypass referential integrity checking.

Files Generated

If you invoke ASNLOAD, the following files are generated:

- <*apply_qual*>.IXF

  This file contains the data exported from the source.

- <*apply_qual*>.EXP

  This file contains error, warning, or informational messages issued by the EXPORT APIs.

- <*apply_qual*>.LOA

  This file contains error, warning, or informational messages issued by the LOAD APIs.

Error Handling

If an error occurs while the Apply program calls ASNLOAD, or if ASNLOAD returns a nonzero return code, the Apply program issues a message, stops processing that subscription, and processes the next subscription.

Returning Control to Users with ASNDONE

If you specify the NOTIFY parameter when starting the Apply program, the user exit program ASNDONE is called after subscription processing completes, regardless of success or failure. You can modify ASNDONE to meet the requirements of your installation. For example, the user exit can examine the UOW table to discover rejected transactions and initiate further actions, such as issuing a message or generating an alert.

See the prolog section in the sample program (ASNDONE.SMP) for instructions on how to modify the program.

Initiating a Forward Recovery with the Apply Program

In cases of an incomplete rollback or partially restored backup, you might need to copy a “window” of changes for completion.

To use the Apply program to limit the range of changes to those excluded from the rollback:

1. Determine how much data is lost.

   Check the IBMSNAP_LOGMARKER value in the point-in-time table. The highest value is the most recently committed change.

2. Use SQL to identify the range of changes. Either:

   - Use the WHERE clause on the Rows page of the Advanced Subscription Definition notebook.

     or
   - Manually update the PREDICATES column of the Common Subscription Targets Member table (ASN.IBMSNAP_SUBS_MEMBER).

   For example, if your timestamp shows the last committed change to be 941106174322, your SQL should reflect changes that occur after that time.

Troubleshooting

The following sections describe commonly known issues with Capture and Apply for OS/2.

Problems Using the Capture Program

Problem

Capture for OS/2 is not capturing updates.

Any of the following could prevent the Capture program from capturing updates:

- Proper authorization was not granted to the user ID running the Capture program. The Capture program requires DBADM or SYSADM privileges.
- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting the Capture and Apply programs was not used:
  1. Define replication sources and subscriptions before starting the Capture program.
  2. Start the Capture program and look for a message number ASN0100I (initialization completed) in the system console or in the ASN.IBMSNAP_TRACE table.
  3. Start the Apply program.

Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

I'm not sure if the Capture program is running successfully.

After you start the Capture and Apply programs, the Apply program performs a full refresh to populate the target tables. Then the Capture program writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which the Capture program starts to capture updates.

Updates captured from then on are placed in change data tables. They are eventually applied to target tables and pruned from the change data table. After the Capture program runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the Capture program's progress. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, the Apply program logs its information in the *userid*.IBMSNAP_APPLYTRAIL table.

Problem

Capture for OS/2 terminated unexpectedly.

The Capture for OS/2 program terminates either because of a severe error, or when you issue the STOP command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0 STOP command issued
8 Error during initialization
12 Any other severe error

Problems Using the Apply Program

Before running the Apply program, ensure that:

- Replication sources and subscriptions have been defined.
- The Capture program has been started, and the ASN0100I initialization message has been issued (if you are running a Capture program).
- Bind has been performed for all control, source, and target servers that the Apply program accesses.

When an error occurs while running the Apply program, the status field in IBMSNAP_SUBS_SET is set to -1 for that copy definition.

Errors as well as successful executions are logged in the IBMSNAP_APPLYTRAIL table. APPERRM contains the message text. Any SQL errors are also logged in this table. Error messages are also issued on the console.

Specific Apply program problems are described below.

Problem

I have performed a successful bind, but when running the Apply program I still get SQLCODE -805, SQLSTATE 51002.

Make sure that the user ID has execute privilege on the Apply packages, and make sure to bind the Capture package to the source server database and to bind both Apply packages to the control, source, and target server databases.

Problem

The DB2 log has filled to capacity because I copied a very large table.

If the error occurred during a full refresh, you can use alternative methods to load large tables. You can either use the ASNLOAD exit, described in Table 5, ASNARUN Invocation Parameter Definitions, or you can perform your own load, described in “Loading Large Copies” on page 138.

If the occurred while applying changed data, then you can change the data blocking parameter to break down large blocks of changed data. See “Specifying Mini-Cycles for the Apply Program to Copy Committed Data” on page 115 in chapter 7.

Problem

The Capture program was cold started, which caused the Apply program to perform a full refresh, but I don't want a full refresh.

If your target table is very large, and in cases where you have decided to use only your own load mechanism, you might want to suppress any future full refreshes of the Apply program. Set the DISABLE_REFRESH flag to 1 in ASN.IBMSNAP_REGISTER at the source server for the source table. In this case, the Apply program issues message ASN1016E.

Problem

A gap was detected, so the Apply program won't perform a full refresh of my target table.

Force a full refresh by resetting the LAST_SUCCESS, SYNCHTIME, and SYNCHPOINT values in ASN.IBMSNAP_SUBS_SET to null.

I unsuccessfully tried to start a second Apply program instance.

You must run each instance with a unique Apply qualifier.

Problem

I received error ASN1003 with SQLCODE = -1032 and SQLSTATE = 57019.

You must start the database manager before invoking the Apply program.

Problem

Apply components for DB2 Universal Database stops with an SQLCODE= -330, SQLSTATE=22517, "A string cannot be used, because its characters cannot be translated".

When copying between DB2 for MVS and DB2 Universal Database, the CCSID translation can cause an INSERT to fail if a translated value is longer than the DB2 column in which it will be inserted. Apply can generate an SQLCODE -330 when it tries to insert a translated COPY_TABLE column value from the refresh control table on a DB2 Universal Database target server into the pruning control table on a DB2 for MVS source server.

For example, if you use the Korean character set with mixed data at both DB2 for MVS source server and DB2 Universal Database target server, the INSERT fails because the original string is in mixed data ASCII. When it is translated to EBCIDIC mixed data with the Korean character set, if the resulting string length is greater than 18 characters (the maximum COPY_TABLE length), the INSERT fails with an SQLCODE of -330.

Important: If you are running in a mixed environment, ensure you have installed the latest maintenance for the CCSID support of your DB2 for MVS program.

For more information on character translation, see the Character Conversion for Distributed Data chapter in *DB2 Version 4 Administration Guide, Volumes 1,2*.

Chapter 13. Capture and Apply for UNIX Platforms

This chapter describes how to set up, operate, and troubleshoot Capture and Apply for the following UNIX platforms:

- AIX
- HP-UX
- Solaris

Read the following sections before reading the sections on operating Capture and Apply.

- "Setting Up the Capture and Apply Programs"
- "Specifying Tuning Parameters for the Capture Program" on page 188
- "Restrictions When Running the Capture Program" on page 189
- "Authorization for Running the Capture Program" on page 189

Setting Up the Capture and Apply Programs

Setting up consists of configuring the source, target, and control servers. The following sections provide instructions for configuring each server as well as information about providing end-user authentication at the source server.

Configuring the Capture Program for UNIX Platforms

1. Ensure that the user ID under which the Capture program is running has the required privileges:
   - Execute privilege on the Capture packages
   - DBADM or SYSADM privileges for the source, control, and target servers
2. Log on with the user ID from the source server system.
3. Connect to the source server database by entering:

   ```
   db2 connect to database
   ```

   where *database* is the source server database.
4. Prepare the source server database (source database) for roll-forward recovery by issuing the **update database configuration** command and the **backup database** command. For example:

   ```
   db2 update database configuration for database_alias using logretain on
   db2 backup database database_alias
   ```

5. Change to the directory where the Capture program bind files are located, which is usually *drive*:\sqllib\bnd.
6. Create and bind the Capture program package to the source server database by entering the following command:

   ```
   db2 bind @capture.lst isolation ur blocking all
   ```

   where *ur* specifies the list in uncommitted read format for greater performance.

If your system does not support UR, substitute CS (cursor stability format) instead. These commands create a list of packages, the names of which can be found in the capture.lst file for your particular platform.

Configuring the Apply Program for UNIX Platforms

1. Ensure that the user ID under which Apply is running has the required privileges:
   - Execute privilege for Apply packages
   - DBADM or SYSADM privileges for the database
2. Log on with the user ID from the target server system.
3. Change to the directory where the Apply program bind files are located, which is usually in *drive*:\sqllib\bnd.
4. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO database
   ```

   where *database* is the source server database.
5. Create and bind the Apply package to the source server database by entering both of the following commands:

   ```
   db2 bind @applycs.lst isolation cs blocking all
   ```

   ```
   db2 bind @applyur.lst isolation ur blocking all
   ```

   Where:

   CS The list in cursor stability format.

   UR The list in uncommitted read format.

   These commands create a list of packages, the names of which can be found in the applycs.lst and applyur.lst files.
6. Connect to the target server database by entering:

   ```
   db2 connect to database
   ```

   where *database* is the target server database.
7. Create and bind the Apply package to the target server database by entering both of the following commands:

   ```
   db2 bind @applycs.lst isolation cs blocking all grant public
   ```

   ```
   db2 bind @applyur.lst isolation ur blocking all grant public
   ```

   Note: Because the Apply control tables use static SQL calls, the Apply bind process searches for nearly all of the control tables at each server that it is bound to, regardless of whether these control tables have any use or meaning at each server.
8. Repeat the connect and bind steps for each server that the Apply program connects to. You must bind the Apply program to the source, target and control servers.

Other Configuration Considerations for UNIX-Based Components

Ensure that the user ID from which the Capture and Apply programs are running have write privilege on the directories from where you invoke the programs. For all other issues pertaining to configuration of UNIX-based components, see *IBM DB2 Universal Database for UNIX Quick Beginnings.*

Providing End-User Authentication at the Source Server

For end-user authentication to occur at the source server, in some cases you need to provide a password file for Apply to use when connecting to the source server. Following are environment-specific changes:

- If you installed Apply for HP-UX, or Apply for Solaris, you must use the AUTH=SERVER scheme and provide a password file.
- If you installed Apply for AIX, you must provide a password file if you want to use the AUTHENTICATION=SERVER scheme. If you use the AUTHENTICATION=CLIENT scheme, you do not need to provide a password file.

For more information about authentication and security, refer to the *IBM DB2 Universal Database Administration Guide.*

If you create a password file:

The password file must meet the following criteria:

- Be named as shown:

  `<APPLYQUAL><instname><CNTLSRVR>.PWD`

  Where:

  APPLYQUAL
  The Apply qualifier in upper case

  instname
  The instance name in lower case that Apply runs under. The value of DB2INSTANCE.

  CNTLSRVR
  The name of the control server in upper case

  For example: `DATADIRapply1REPRTDB.PWD`

  Note that this naming convention is the same as the log file name (.LOG) and the spill file name (.SPL), but with a file extension of PWD.

- Reside in the directory from which the Apply program starts
- Contain all server-name/password pairs for the file. This enables you to use a different (or the same) password at each server. The Apply user ID is used for all connections.
- Have one or more records using the following format:

  `SERVER=server_name PWD=password`

The file cannot include blank lines or comment lines.

It is advisable to limit read access of this file to the user ID that will run Apply.

If you do not create a password file:

Apply for UNIX-based platforms must be able to issue an SQL CONNECT statement without specifying the user ID and password. To do so, ensure that:

- The DB2 for MVS database is catalogued as AUTHENTICATION=CLIENT.
- The login ID belongs to PRIMARY GROUP=SYSTEM.

When copying from DB2 for MVS sources, ensure that:

- SECURITY=SAME for MVS CPI-C node.
- You specify the following values when you define the LU name via the VTAM APPL:
  - VERIFY=NONE to indicate that any LU can request an LU-LU session.
  - SECACPT=ALREADYV to indicate user ID and password checking at the requester.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit
The number of minutes to keep the change data table rows and the unit-of-work (UOW) table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit
The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If Capture shuts itself down, you should perform a cold start if the database does not have or support an archive log.

Commit interval
The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if Capture and Apply are running at the same time. This precaution helps to avoid locking overhead. If Apply is *not* running at the same time as the Capture program, you can set the commit interval no higher than the DB2 timeout interval.

Prune interval
The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is ignored if you start Capture with the NOPRUNE option.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

```
UPDATE TABLE ASN.IBMSNAP_CCPPARMS
SET RETENTION_LIMIT=number_of_minutes,
LAG_LIMIT=number_of_minutes,
COMMIT_INTERVAL=number_of_seconds,
PRUNE_INTERVAL=number_of_seconds
```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.
- Drop a replication source table.
- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Other Capture program restrictions are:

- Capture does not capture any changes made by DB2 utilities.
- Capture for AIX does not support large objects and LONG data types.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have DBADM or SYSADM authority.

Operating Capture for AIX, Capture for HP-UX, and Capture for Solaris

The IBM Replication administrator user ID can use the commands in this section to operate Capture for UNIX platforms. Enter the commands or key combinations from an AIX, HP-UX, or Solaris window.

This section explains how to perform the following Capture tasks:

- Starting
- Scheduling
- Stopping

- Suspending
- Resuming
- Reinitializing
- Pruning

Before You Start the Capture Program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where Capture is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS
  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC

  You can also create these control tables manually by running the DPCNTL.* file from the RUN SQL Files window.
- Bind the Capture program to the source servers from which the Capture program will capture changes. See “Configuring the Capture Program for UNIX Platforms” on page 185 for more information.
- Enable the DATA CAPTURE CHANGES attribute for your source table using either the CREATE TABLE or ALTER TABLE statement.

Starting Capture for AIX, Capture for HP-UX, and Capture for Solaris

After you start Capture, it runs continuously until you stop it or it detects an error.

To start Capture for AIX, Capture for HP-UX, and Capture for Solaris:

1. From AIX, HP-UX, or Solaris, log in and make sure that the user ID under which Capture is running has write privilege on the directory.
2. Ensure that you set the DB2 instance as shown:

   ```
   export DB2INSTANCE=db2_instance_name
   ```

   While the Capture program is running, a file with the name <*database_instance_name*><*database_name*>.CCP is created in the directory from which the Capture program is invoked. This file is a log file for the messages issued by the Capture program; these messages are also recorded in the ASN.IBMSNAP_TRACE table.
3. Enter the following command:

```
►►──asnccp──┬────────────┬──┬────────┬──┬─────────┬──┬──────────┬──────────►◄
            └─src_server─┘  ├─warm───┤  ├─prune───┤  ├─notrace──┤
                            ├─warmns─┤  └─noprune─┘  └─trace────┘
                            └─cold───┘
```

Table 15 defines the optional parameters.

Table 15. ASNCCP Command Parameter Definitions for UNIX Platforms

Parameter	Definition
src_server	Source server name positional parameter that must be the first parameter if entered. If not specified, the value from the DB2DBDFT environment variable will be used.
warm (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. See “Warm and Cold Starts” on page 195 for more information.
warmns	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With warmns, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, Capture terminates instead of switching to a cold start as when warm is specified. See “Warm and Cold Starts” on page 195 for more information.
cold	The Capture program starts by deleting all rows in its change data table, UOW table, pruning control table, and trace table during initialization. Refer to “Warm and Cold Starts” on page 195 for more information.
prune (default)	The Capture program automatically prunes the change data and UOW tables.
noprune	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the prune command. See the PRUNE command for more information.
notrace (default)	No trace information is written.
trace	Writes debug trace messages to the standard output, *stdout.*

Scheduling Capture for AIX, Capture for HP-UX, and Capture for Solaris

Use the **at** command to start Capture at a specific time. For example, the following command starts Capture at 3:00 p.m. on Friday:

```
at 3pm Friday asnccp warmns noprune
```

Stopping Capture for AIX, Capture for HP-UX, and Capture for Solaris

```
►►──asncmd──stop──────────────────────────────────────────────────────────►◄
```

or

```
►►──kill──-INT──pid──────────────────────────────────────────────────────────►◄
```

or one of the following:

- Ctrl+C
- Ctrl+D

Purpose

Use the **stop** command, the **kill** command, or the key combinations to stop the Capture program in an orderly way and commit the log records that it processed up to that point.

Stop Capture before:

- Removing an existing replication source
- Opening an existing replication source and modifying the **Enable Refresh and Update** field

Usage

- If you want to use the **stop** command, do the following from a different window than where Capture is running:
  1. Export DB2INSTANCE to the value that was set when the Capture program was started.
  2. Export DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
  3. Enter **asncmd stop**.
- If you want to use the **kill** command, find out the process id of the Capture program and enter **kill -int pid**. (You do not need to set the environment variables.)

Suspending Capture for AIX, Capture for HP-UX, and Capture for Solaris

```
►►──asncmd──suspend──────────────────────────────────────────────────────────►◄
```

Purpose

Use the **suspend** command to relinquish AIX, HP-UX, or Solaris resources to operational transactions during peak periods without damaging the Capture program environment.

This command suspends the Capture program until you issue the **resume** command.

Attention:

Do not use the **suspend** command when canceling a replication source. Instead, stop the Capture program.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Resuming Capture for AIX, Capture for HP-UX, and Capture for Solaris

```
►►──asncmd──resume─────────────────────────────────────────────────►◄
```

Purpose

Use the **resume** command to restart the Capture program if you suspended it using the **suspend** command.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Reinitializing Capture for AIX, Capture for HP-UX, and Capture for Solaris

```
►►──asncmd──reinit─────────────────────────────────────────────────►◄
```

Purpose

Use the **reinit** command to have the Capture program reread the ASN.IBMSNAP_REGISTER tables or the ASN.IBMSNAP_CCPPARMS table while it is running. Use this command when you add a new replication source with Control Center while the Capture program is running.

This command ensures that the Capture program recognizes the new replication source. The Capture program also rereads the ASN.IBMSNAP_CCPPARMS table for any changes made to the tuning parameters.

Attention:

Do not use the **reinit** command to reinitialize the Capture program after canceling a replication source or dropping a replication source table while Capture is running. Instead, stop the Capture program and restart it using the WARM option.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Pruning the Change Data and Unit-of-Work Tables

```
►►──asncmd──prune──────────────────────────────────────────────────►◄
```

Purpose

Use the **prune** command to perform pruning of the change data (CD) table and UOW table if you used the **noprune** invocation parameter.

This command prunes the CD and UOW tables once if pruning was disabled by using the Noprune parameter while starting the Capture program. The Capture program issues the message ASN0124I when the command is successfully queued.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

During pruning, if you stop or suspend Capture, pruning does not resume after you enter the **resume** command. You must enter the **prune** command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──asncmd──getlseq──────────────────────────────────────────────►◄
```

Purpose

Use the **getlseq** command to provide the current log sequence number and the timestamp. You can use this number to determine how far Capture has read the DB2 log.

Usage

To use the command, do the following from a different window than where the Capture program is running:

1. Set environment variable DB2INSTANCE to the value that was set when the Capture program was started.
2. Set environment variable DB2DBDFT to the source server specified when the Capture program was started (or the DB2DBDFT value that was used when the Capture program was started).
3. Enter the command.

Warm and Cold Starts

This section explains how Capture handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the warm start table, ASN.IBMSNAP_WARM_START, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, UOW table, or change data tables.

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, an operator might cancel the Capture program or stop DB2. In this case, the Capture program uses the change data, UOW, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (Apply performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued. For information about handling gap messages, see “Problems Using the Apply Program” on page 202 and read the portion about forcing Apply to perform a full refresh.)

Automatic Cold Starts

Sometimes the Capture program automatically switches to a cold start, even if you specified a warm start (not WARMNS). For example:

- The warm start log sequence lags behind the current log sequence by more than the LAG_LIMIT value (as specified in the tuning parameters table ASN.IBMSNAP_CCPPARMS).
- You invoke the Capture program for the first time.

  The first time that you invoke the Capture program, you see message ASN0102W, indicating that the warm start failed. Capture switches to a cold start. You can ignore this message when first invoking the Capture program.

For each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, or if someone brings down the DB2 table space containing the change data table. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Operating Apply for AIX, Apply for HP-UX, and Apply for Solaris

The IBM Replication administrator can use the commands in the following sections to perform the following Apply for Windows NT tasks:

- Starting
- Scheduling
- Stopping

Before You Start the Apply Program

Before you start the Apply program, ensure that:

- The administrator created the control tables.
- A password file has been created, if necessary, for end-user authentication at any server. See “Providing End-User Authentication at the Source Server” on page 187 for more information.
- At least one subscription is created and activated.
- The Apply package is created. See “Configuring the Apply Program for UNIX Platforms” on page 186 for information on bind programs to create Apply packages for UNIX platforms.
- You have the proper authorization. See “Authorization Requirements for the Capture Program” on page 66 for information about authorization for the Apply program.

Starting Apply for AIX, Apply for HP-UX, and Apply for Solaris

After you start Apply, it runs continuously until:

- You stop Apply in an orderly way.
- An unexpected error or failure occurs.

To start Apply for AIX, Apply for HP-UX, and Apply for Solaris:

1. Ensure that you set the DB2 instance name as shown:

   ```
   export DB2INSTANCE=db2_instance_name
   ```

2. From AIX, HP-UX, or Solaris, log on with the IBM Replication user ID. Make sure that the user ID under which Apply is running has write privilege on the directory.
3. Enter the **asnapply** command and one or more options, if desired.

```
►►──asnapply──apl_qual──┬───────────┬──┬───────────┬──┬──────────┬──────►
                        └─ctrl_serv─┘  ├─loadxit───┤  ├─inamsg───┤
                                       └─noloadxit─┘  └─noinamsg─┘

►──┬─────────┬──┬──────────┬────────────────────────────────────────────►◄
   ├─notrc───┤  ├─notify───┤
   ├─trcerr──┤  └─nonotify─┘
   └─trcflow─┘
```

Table 16 on page 198 defines the optional positional parameters.

Table 16. ASNAPPLY Parameter Definitions for UNIX Platforms

Parameter	Definition
apl_qual	Specifies the Apply qualifier that the Apply instance uses to identify the subscriptions to be served. This positional parameter is required as the first parameter.
ctrl_serv	Specifies the name of the server where the replication control tables will reside. If you do not specify this optional positional parameter, the default is the default database or the value you specified in DB2DBDFT.
loadxit	Specifies that Apply is to invoke ASNLOAD, an exit program that can call an IBM or multivendor utility, to initialize a target table. This keyword parameter is optional.
noloadxit (default)	Specifies that Apply will not invoke ASNLOAD.
inamsg (default)	Specifies that Apply is to issue a message when Apply is inactive.
noinamsg	Specifies that Apply will not issue this message. This keyword parameter is optional.
notrc (default)	Specifies that Apply does not generate a trace.
trcerr	Specifies that Apply generates a trace that contains only error information. This keyword parameter is optional.
trcflow	Specifies that Apply generates a trace that contains error and execution flow information. This keyword parameter is optional.
notify	Specifies that Apply is to invoke ASNDONE, an exit program that returns control to the user when Apply processing ends. This keyword parameter is optional.
nonotify (default)	Specifies that Apply will not invoke ASNDONE.

When Apply for AIX, Apply for HP-UX, or Apply for Solaris executes, it generates two files:

- *<userid><database_instance_name><database_name>*.LOG

  This file contains the log messages issued by Apply during execution. Apply also logs messages to the audit trail table. Check the columns APPERRM, SQLSTATE, SQLCODE, SQLERRP, and SQLERRM for information.

- *<userid><database_instance_name><database_name>*.SPL

  This is an internal file used by Apply.

Scheduling Apply for AIX, Apply for HP-UX, and Apply for Solaris

Use the **at** command to start Apply at a specific time. For example, the following command starts Apply at 3:00 p.m. on Friday:

```
at 3pm Friday asnapply l 1 r
```

Stopping Apply for AIX, Apply for HP-UX, and Apply for Solaris

Use the following command to stop Apply for AIX, Apply for HP-UX, and Apply for Solaris.

```
►►──asnastop──apply_qualifier──────────────────────────────────────►◄
```

or

```
►►──kill──-INT──pid──────────────────────────────────────────────►◄
```

You can alternatively use one of the following key combinations from the window where the Apply program is running:

- Ctrl+C
- Ctrl+Break

Purpose

Use the command or the key combination to stop the Apply program in an orderly way and commit the log records that it processed up to that point.

Usage

- To use the command, do the following from a different window than where the Apply program is running:
  1. Set environment variable DB2INSTANCE to the value that was set when the Apply program was started.
  2. Set environment variable DB2DBDFT to the source server specified when the Apply program was started (or the DB2DBDFT value that was used when the Apply program was started).
  3. Enter the command.
- To use one of the key combinations, press it from the same NT window in which the Apply program is running.

Additional Apply Operations

The following section provides information about performing the additional Apply program functions of scheduling subscriptions, refreshing target tables, returning control to users, and initiating a forward recovery.

Scheduling Subscriptions with the Event Table

You can include an event name in your replication subscription. Your subscription cycles can run according to a timer, an event occurrence, or both. See “Setting the Copying Schedule: Time or Event Based” on page 113 for more information about the purpose and usage of event scheduling.

You can schedule a subscription by manipulating the event table. The DDL statement for the event table appears as follows:

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_EVENT (
  EVENT_NAME CHAR(18) NOT NULL,
  EVENT_TIME TIMESTAMP NOT NULL,
  END_OF_PERIOD TIMESTAMP);
```

For example, to start Apply at 6:00 p.m. on 9/24/97 and copy changes from the source server up to 4:00 p.m. on the same date, you would insert the following values in ASN.IBMSNAP_SUBS_EVENT:

```
('EVENTA','1997-09-24-18.00.00.00000','1997-09-24-16.00.00.00000'),
```

A subscription becomes eligible for processing when EVENT_TIME falls between LASTSUCCESS and the control server's CURRENT TIMESTAMP. The END_OF_PERIOD column enables you to specify a timestamp upper limit for change data.

Refreshing Target Tables with ASNLOAD

ASNLOAD is a user exit program that Apply invokes when you specify the LOADX invocation parameter. Apply calls the ASNLOAD program whenever it performs a full refresh of a target table.

You can use the ASNLOAD program as shipped with the Apply program, or you can modify it. As shipped, ASNLOAD uses the EXPORT utility to export data from the source table and uses the LOAD utility to fully refresh the target table.

You can modify ASNLOAD to call any IBM or vendor utility. See the prolog section in the sample program (ASNLOAD.smp) for instructions on how to modify the program to meet the requirements of your installation.

You must use ASNLOAD to fully refresh tables with referential integrity constraints in order to bypass referential integrity checking.

Files Generated

If you invoke ASNLOAD, the following files are generated:

- ASNA<*userid*><*database_instance_name*><*database_name*>.IXF

  This file contains the data exported from the source.

- ASNAEXPT<*userid*><*database_instance_name*><*database_name*>.MSG

  This file contains error, warning, or informational messages issued by the EXPORT APIs.

- ASNAIMPT<*userid*><*database_instance_name*><*database_name*>.MSG

  This file contains error, warning, or informational messages issued by the LOAD APIs.

Error Handling

If an error occurs while Apply calls ASNLOAD, or if ASNLOAD returns a nonzero return code, Apply issues a message, stops processing that subscription, and processes the next subscription.

Returning Control to Users with ASNDONE

If you specify the NOTIFY parameter when starting the Apply program, the user exit program ASNDONE is called after subscription processing completes, regardless of success or failure. You can modify ASNDONE to meet the requirements of your installation. For example, the user exit can examine the UOW table to discover rejected transactions and initiate further actions, such as issuing a message or generating an alert.

See the prolog section in the sample program (ASNDONE.smp) for instructions on how to modify the program.

Initiating a Forward Recovery with the Apply Program

In cases of an incomplete rollback or partially restored backup, you might need to copy a “window” of changes for completion.

To use Apply to limit the range of changes to those excluded from the rollback:

1. Determine how much data is lost.

   Check the IBMSNAP_LOGMARKER value in the point-in-time table. The highest value is the most recently committed change.

2. Use SQL to identify the range of changes. Either:

   - Use the WHERE clause on the Rows page of the Advanced Subscription Definition notebook.

     or

   - Manually update the PREDICATES column of the Common Subscription Targets Member table (ASN.IBMSNAP_SUBS_MEMBER).

   For example, if your timestamp shows the last committed change to be 941106174322, your SQL should reflect changes that occur after that time.

Troubleshooting

The following sections describe commonly known issues with Capture and Apply for UNIX platforms.

Problems Using the Capture Program

Problem

The Capture program is not capturing updates.

Any of the following could prevent the Capture program from capturing updates:

- Proper authorization was not granted to the user ID running the Capture program. Capture requires DBADM or SYSADM privileges.
- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting Capture and Apply was not used:

1. Define replication sources and subscriptions before starting the Capture program.
2. Start the Capture program and look for a message number ASN0100I (initialization completed) in the system console or in the ASN.IBMSNAP_TRACE table.
3. Start Apply.

Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

I'm not sure if the Capture program is running successfully.

Once Capture and Apply are started, Apply performs a full refresh to populate the target tables. Then Capture writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which Capture starts to capture updates.

Updates captured from then on are placed in change data tables. They are eventually applied to target tables and pruned from the change data table. After Capture runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the progress made by Capture. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, Apply logs its information in the *userid*.IBMSNAP_APPLYTRAIL table.

Problem

The Capture program terminated unexpectedly.

The Capture program terminates either because of a severe error, or when you issue the **stop** command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0	**stop** command issued
8	Error during initialization
12	Any other severe error

Problems Using the Apply Program

Before running Apply, ensure that:

- Replication sources and subscriptions have been defined.
- The Capture program has been started, and the ASN0100I initialization message has been issued (if you are running a Capture program).
- Bind has been performed for all control, source, and target servers that Apply accesses.

When an error occurs while running Apply, the status field in IBMSNAP_SUBS_SET is set to -1 for that copy definition.

Errors as well as successful executions are logged in the IBMSNAP_APPLYTRAIL table. APPERRM contains the message text. Any SQL errors are also logged in this table. Error messages are also issued on the console.

Specific Apply problems are described below.

Problem

I have performed a successful bind, but when running Apply I still get SQLCODE -805, SQLSTATE 51002.

Make sure that the user ID has execute privilege on the Apply packages, and make sure to bind the Capture package to the source server database and to bind both Apply packages to the control, source, and target server databases.

Problem

The DB2 log has filled to capacity because I copied a very large table.

If the error occurred during a full refresh, you can use alternative methods to load large tables. You can either use the ASNLOAD exit, described in Table 5, ASNARUN Invocation Parameter Definitions, or you can perform your own load, described in “Loading Large Copies” on page 138.

If the error occurred while applying changed data, you can change the data blocking parameter to break down large blocks of changed data. See “Specifying Mini-Cycles for the Apply Program to Copy Committed Data” on page 115 in chapter 7.

Problem

Capture was cold started, which caused Apply to perform a full refresh, but I don't want a full refresh.

If your target table is very large, and in cases where you have decided to use only your own load mechanism, you might want to suppress any future full refreshes of Apply. Set the DISABLE_REFRESH flag to 1 in ASN.IBMSNAP_REGISTER at the source server for the source table. In this case, Apply issues message ASN1016E.

Problem

A gap was detected, so Apply won't perform a full refresh of my target table.

Force a full refresh by resetting the LAST_SUCCESS, SYNCHTIME, and SYNCHPOINT values in ASN.IBMSNAP_SUBS_SET to null.

I unsuccessfully tried to start a second Apply instance.

You must run each instance with a unique Apply qualifier.

Problem

I received error ASN1003 with SQLCODE = -1032 and SQLSTATE = 57019.

You must start the database manager before invoking Apply.

Problem

Apply components for DB2 Universal Database stops with an SQLCODE= -330, SQLSTATE=22517, "A string cannot be used, because its characters cannot be translated".

When copying between DB2 for MVS and DB2 Universal Database, the CCSID translation can cause an INSERT to fail if a translated value is longer than the DB2 column in which it will be inserted. Apply can generate an SQLCODE -330 when it tries to insert a translated TARGET_TABLE column value from the refresh control table on a DB2 Universal Database copy server into the pruning control table on a DB2 for MVS source server.

For example, if you use the Korean character set with mixed data at both DB2 for MVS source server and DB2 Universal Database target server, the INSERT fails because the original string is in mixed data ASCII. When it is translated to EBCIDIC mixed data with the Korean character set, if the resulting string length is greater than 18 characters (the maximum TARGET_TABLE length), the INSERT fails with an SQLCODE of -330.

Important: If you are running in a mixed environment, ensure you have installed the latest maintenance for the CCSID support of your DB2 for MVS program.

For more information on character translation, see the Character Conversion for Distributed Data chapter in *DB2 Version 4 Administration Guide, Volumes 1,2*.

Chapter 14. Capture for VSE

This chapter describes how to set up, operate, and troubleshoot Capture for VSE.

Read the following sections before reading the sections on operating Capture for VSE:

- “Setting Up the Capture Program”
- “Specifying Tuning Parameters for the Capture Program”
- “Restrictions When Running the Capture Program” on page 206
- “Authorization for Running the Capture Program” on page 207
- “Recovering from Severe Errors” on page 207

Setting Up the Capture Program

Setting up consists of installing the Capture for VSE program and configuring the source servers. All Capture for VSE program required tables must be defined prior to invoking the Capture program. These are created by the Control Center when a replication source is defined. You can also create these control tables manually by running the DPCNTL.* file from the RUN SQL Files window.

See the Capture for VSE program directory for instructions installing the Capture program.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit
The number of minutes to keep the change data table rows and the unit-of-work (UOW) table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit
The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If the Capture program shuts itself down, you should perform a cold start if the database log has been archived and the log has wrapped around.

Commit interval
The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if the Capture and Apply programs are running at the same time. If the Apply program is *not* running at the same time as Capture program, you can set the commit interval no higher than the DB2 timeout interval.

Prune interval
The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is

ignored if you start Capture program with the NOPRUNE option; however, you can override this option with the PRUNE command.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

```
UPDATE TABLE ASN.IBMSNAP_CCPPARMS
SET RETENTION_LIMIT=number_of_minutes,
LAG_LIMIT=number_of_minutes,
COMMIT_INTERVAL=number_of_seconds,
PRUNE_INTERVAL=number_of_seconds,
```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.
- Drop a replication source table.
- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Other Capture program restrictions are:

- Changes are not captured when the database is running with LOGMODE=N.
- One change data table exists per user table defined to the Capture program.
- You must specify DATA CAPTURE CHANGES on the CREATE or ALTER TABLE statement for the user tables.
- Tables with field procedures for columns (FIELDPROC specified on CREATE or ALTER TABLE) are not supported by Capture for VSE.
- There can be only one Capture program running per DB2 server for VSE database, and each Capture program runs in its own partition.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have DBA authority.

When running Capture for VSE with the DB2 server for VSE, the Capture program job also dynamically creates a table to determine whether or not a Capture program is already running.

Recovering from Severe Errors

Capture for VSE provides you with the following tools to assist you if a severe error occurs:

- Alert generation
- Trace buffer
- Trace output
- Storage dump

The following sections provide information on these tools.

Trace Buffer

Capture for VSE puts a small amount of critical diagnostic data in a wraparound trace buffer during processing. Each trace buffer entry describes current data capture status. If a severe error occurs, Capture prints the trace buffer before termination. The printing of the trace buffer supplements the Capture program error message.

Trace Output

When an error occurs, you can run the Capture program with the TRACE option. When you use this option, IBM Replication writes trace information logic flow to STDOUT.

Storage Dump

When Capture for VSE terminates with a severe error, it saves critical diagnostic data in the CEEDUMP data set. This information is more detailed than the information in the Capture program trace buffer.

Operating Capture for VSE

The replication administrator can use the commands in this section to operate Capture for VSE.

This section explains how to perform the following Capture for VSE tasks:

- Starting
- Stopping
- Suspending
- Resuming
- Reinitializing
- Pruning

Before You Start the Capture Program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where the Capture program is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS
  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC
- Ensure that the administrator has DBA privilege.
- Ensure that ASN.IBMSNAP_REGISTER has at least one entry in it by defining a replication source with the DATA CAPTURE CHANGES option.
- Enable the DATA CAPTURE CHANGES attribute for your source tables using either the CREATE TABLE or ALTER TABLE statement.

The ASNS51CD job control member contains the name of the messages file used. The default is American English. Modify ASNS51CD to issue messages in a different language. See the Capture program directory for a list of supported languages.

Starting Capture for VSE

After it starts, the Capture program runs continuously until you stop it or it detects an error.

To start the Capture program:

Start it in a partition like a batch job. Sample job control member ASNS51BD provides an example of how to start the Capture program. You can specify ASNCCP invocation parameters in the PARM field, in the order shown, separated by one or more blanks:

```
►►──//──EXEC──PGM=ASNCCP──,PARM=──'──┬────────┬──┬─────────┬──────────────►
                                      ├─WARM───┤  ├─PRUNE───┤
                                      ├─COLD───┤  └─NOPRUNE─┘
                                      └─WARMNS─┘

►──┬─────────┬──┬──────────┬──┤ Group 1 ├──'──────────────────────────►◄
   ├─TERM────┤  ├─TRACE────┤
   └─NOTERM──┘  └─NOTRACE──┘
```

Group 1:

```
├──USERID=──auth_name/password──┬─────────────────────────┬──────────────┤
                                └─Dbname=──databasename───┘
```

Table 17 on page 209 defines the optional parameters.

Table 17. ASNCCP Command Parameter Definitions for VSE

Parameter	Definition
Src_server	Source server name positional parameter that must be the first parameter if entered. If not specified, the value from the DB2DBDFT environment variable will be used.
WARM (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. See “Warm and Cold Starts” on page 156 for more information.
WARMNS	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With WARMNS, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, the Capture program terminates instead of switching to a cold start as when WARM is specified. See “Warm and Cold Starts” on page 156 for more information.
COLD	The Capture program starts by deleting all rows in its change data, UOW, pruning control, and trace tables during initialization. See “Warm and Cold Starts” on page 156 for more information.
PRUNE (default)	The Capture program automatically prunes the change data and UOW tables.
NOPRUNE	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the PRUNE command. See the PRUNE command for more information.
NOTRACE (default)	No trace information is written.
TRACE	Writes debug trace messages to the standard output, *stdout*.
USERID=*auth_name/password*	Specifies that the Capture program should connect to the database as user ID *auth_name*. The correct password must be provided or an error is returned. The auth_name and password are both from 1 to 8 characters in length.
Dbname=*database_name*	Identifies the name of the DB2 server for VSE database for which changes are to be captured. It is from 1 to 18 characters in length. If not specified, the default is the database name as specified in the DBNAME directory or SQLDS if a DBNAME directory has not been set up.

Stopping Capture for VSE

```
►►──MSG──partition,DATA=STOP──────────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Use the STOP command to stop the Capture program gracefully and commit the log records that it processed up to that point.

Stop Capture before:

- Removing an existing replication source
- Opening and modifying an existing replication source
- Shutting down the database

Usage

If you stop the Capture program, the Capture program shuts itself down and issues an informational message. If it detects an error, the program shuts itself down after cleaning up the data in the affected tables (so that the data will not be used). Staging tables are pruned when it is appropriate. In the case of re-IPL or logoff, you must initiate a cold start because the warm start information could not be saved.

Capture for VSE issues informational or error messages to the console.

Suspending Capture for VSE

```
►►──MSG──partition,DATA=SUSPEND──────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Use the SUSPEND command to suspend the Capture program until you issue the RESUME command.

Usage

You can use this command to suspend the Capture program to improve performance for operational transactions during peak periods without destroying the Capture for VSE program run environment.

Resuming Capture for VSE

```
►►──MSG──partition,DATA=RESUME───────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Use the RESUME command to resume the suspended Capture program.

Usage

You can use this command to resume the Capture program if you suspended it with the SUSPEND command.

Reinitializing Capture for VSE

```
►►──MSG──partition,DATA=REINIT────────────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Use the REINIT command to reinitialize the Capture program.

Usage

Use the REINIT command to begin to capture changes from new source tables if you add a new replication source with the Control Center while the Capture program is running. This command ensures that the Capture program recognizes the new replication sources in the change data control table. To allow the new replication source to be visible to the Capture program, you must issue a REINIT command. The REINIT command tells the Capture program to obtain newly added replication sources from the change data control table (ASN.IBMSNAP_REGISTER).

REINIT also rereads the tuning parameters table (ASN.IBMSNAP_CCPARMS) for any changes made to the tuning parameters.

Attention: Do not use REINIT to reinitialize the Capture program after canceling a replication source or dropping a replication source table while the Capture program is running. Instead, stop the Capture program and use warm start.

Pruning the Change Data and Unit-of-Work Tables

```
►►──MSG──partition,DATA=PRUNE─────────────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Use the PRUNE command to initiate the pruning of the change data (CD) and unit-of-work (UOW) tables, if you used the NOPRUNE invocation parameter on ASNCCP.

Usage

This command prunes tables once. During pruning, if you stop or suspend the Capture program, pruning does not resume after you enter the RESUME command. You must enter the PRUNE command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──MSG──partition,DATA=GETLSEQ───────────────────────────────────────►◄
```

where *partition* represents the partition that is running Capture for VSE.

Purpose

Provides the timestamp and current log sequence number.

Usage

Use the GETLSEQ command to determine how far the Capture program has read the log.

Warm and Cold Starts

This section explains how the Capture program handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the warm start table, ASN.IBMSNAP_WARM_START, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, unit-of-work table, or change data tables.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (The Apply program performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued.

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, you might cancel the Capture program or stop DB2 with SQLEND QUICK in a VSE environment. In this case, the Capture program uses the change data, unit-of-work, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

Automatic Cold Starts

Sometimes the Capture program automatically switches to cold start, even when you specify a warm start. (However, the switch to a cold start is not made when WARMNS is specified.) The switch is made when:

- The warm start log sequence lags behind the current log sequence by more than the LAG_TIME value as specified in the tuning parameters table, ASN.IBMSNAP_CCPPARMS.
- No warm start information is available, because it could not be saved.
- The warm start log sequence is not available on the DB2 server for VSE active log. For example, this could happen if a COLDLOG was performed or a restore from an older archive was performed.

- The Capture program cannot keep up with the amount of log activity, and the log wraps.
- You invoke the Capture program for the first time.

In each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, and the NOTERM parameter was not specified. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Troubleshooting: Problems Using the Capture Program

This section describes commonly known issues with the Capture program. Before running Capture for VSE, ensure that you have all the PTFs for your level of DB2. Specific Capture program problems are described below.

Problem

Capture for VSE does not start.

Ensure that:

- Access has been given for the database log and directory minidisks by specifying the DLBLs in the Capture program: startup JCL.
- Access has been given to the C Run Time Library.
- The ASNLMAIN package file has been loaded to the database.

Problem

The Capture program is not capturing updates.

Any of the following could prevent Capture for VSE from capturing updates:

- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting the Capture and Apply programs was not used:
    1. Replication sources and subscriptions must be defined before starting Capture for VM.
    2. Start Capture for VM and look for message number ASN0100I (initialization completed) on the virtual machine console or in the ASN.IBMSNAP_TRACE table.
    3. Start the Apply program.

Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

I'm not sure if the Capture program is running successfully.

After you start the Capture and Apply programs, the Apply program performs a full refresh to populate the target tables. Then the Capture program writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which the Capture program starts to capture updates.

Subsequent captured updates are placed in change data tables and eventually applied to target tables and pruned from the change data table. After the Capture program runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the progress made by the Capture program. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, the Apply program logs its information in the *userid*.IBMSNAP_TRAIL table.

Problem

Capture for VSE: issued message ASN0000E instead of the proper message number.

Capture for VSE issued a generic message instead of a proper message. This happens if the specified VSAM message file in the Capture startup JCL was not found. See the Capture for VSE program directory for information on installing the VSAM message file.

Problem

Capture for VSE terminated unexpectedly.

The Capture for VSE program terminates either because of a severe error, or when you issue the STOP command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0	STOP command issued
8	Error during initialization
12	Any other severe error

Chapter 15. Capture for VM

This chapter describes how to set up, operate, and troubleshoot Capture for VM.

Read the following sections before reading the sections on operating Capture for VM:

- “Setting Up the Capture Program”
- “Specifying Tuning Parameters for the Capture Program”
- “Restrictions When Running the Capture Program” on page 216
- “Authorization for Running the Capture Program” on page 217
- “Recovering from Severe Errors” on page 217

Setting Up the Capture Program

Setting up consists of installing the Capture for VM program and configuring the source servers. All required Capture for VM program tables must be defined prior to invoking the Capture program. These are created by the DB2 Version 5 Control Center when a replication source is defined. You can also create these control tables manually by running the DPCNTL.* file from the Run SQL File window.

See the Capture for VM program directory for instructions about installing the Capture program.

Specifying Tuning Parameters for the Capture Program

To control the performance of the Capture program, you can specify the following tuning parameters in the ASN.IBMSNAP_CCPPARMS tuning parameters table:

Retention limit
The number of minutes to keep the change data table rows and the unit-of-work (UOW) table rows. The default value is 10,800, which is 7 days. The rows are deleted up to where the changes have been applied.

Lag limit
The number of minutes the Capture program can be backlogged from the current local time before shutting itself down. The default value is 10,800 (which is 7 days). This value is higher for a busy system; therefore, a lower lag limit shuts down the Capture program. If the Capture program shuts itself down, you should perform a cold start if the database log has been archived and the log has wrapped around.

Commit interval
The number of seconds to wait before issuing a COMMIT statement. The default value is 30 seconds. Set the interval smaller than the DB2 timeout interval if the Capture and Apply programs are running at the same time. If the Apply program is *not* running at the same time as the Capture program, you can set the commit no higher than the DB2 timeout interval.

Prune interval
The number of seconds to wait before pruning the staging tables. The default value is ten times the commit value or 600 seconds, whichever is larger. This parameter is

ignored if you start the Capture program with the NOPRUNE option; however, you can override this option with the PRUNE command.

To specify the tuning parameters, do one of the following tasks:

- Modify DPCNTL.* in the Control Center /sqllib/bin directory before you define the first replication source for a database.
- Update the table with the following SQL statement after you create the tuning parameters table:

  ```
  UPDATE TABLE ASN.IBMSNAP_CCPPARMS
  SET RETENTION_LIMIT=
  number_of_minutes,
  LAG_LIMIT=
  number_of_minutes,
  COMMIT_INTERVAL=
  number_of_seconds,
  PRUNE_INTERVAL=
  number_of_seconds,
  ```

If you need to change the values and refresh the tuning parameters while the Capture program is running, enter the REINIT command after changing the table values.

For information on the structure of the tuning parameters table, see Chapter 20, “Table Structures” on page 265.

Restrictions When Running the Capture Program

The following actions cause the Capture program to terminate while it is running. Stop the Capture program if you want to perform any of the following tasks:

- Cancel an existing replication source.
- Drop a replication source table.
- Make changes that affect the structure of source tables. This includes changes resulting from data definition language or utilities. Structural changes can compromise the data integrity of the copies.

Other Capture program restrictions are:

- Changes are not captured when the database is running with LOGMODE=N.
- One change data table exists per user table defined to the Capture program.
- You must specify DATA CAPTURE CHANGES on the CREATE or ALTER TABLE statement for the user tables.
- Tables with field procedures for columns (FIELDPROC specified on CREATE or ALTER TABLE) are not supported by Capture for VM.
- There can be only one Capture program per database, and each Capture program runs in its own virtual machine. The Capture program identifies itself as an APPC/VM resource. By default, the resource ID value is CAPTURE. To use a dif-

ferent resource ID or to allow multiple Capture programs to run on the system (for different DB2 databases), change the ENQ_NAME parameter in the ASNPARMS file.

- Because the Capture program identifies itself as an APPC/VM resource, you must specify appropriate IUCV VM/ESA System Directory control statements (such as IUCV *IDENT RESANY GLOBAL) for virtual machines that run the Capture program. For more information, see the *VM/ESA Planning and Administration Guide*.
- The Capture program requires access to the appropriate level of the C Run Time Library. On VM, you must issue GLOBAL LOADLIB SCEERUN before invoking ASNCCP.

Authorization for Running the Capture Program

The user ID that operates the Capture program should either be the replication administrator or have DBA authority.

When running the Capture for VM with the DB2 server for VM, the Capture program job also dynamically creates a table to determine whether or not a Capture program is already running.

Recovering from Severe Errors

Capture for VM provides you with the following tools to assist you if a severe error occurs:

- Trace buffer
- Trace output
- Storage dump

The following sections provide information on these tools.

Trace Buffer

Capture for VM puts a small amount of critical diagnostic data in a wraparound trace buffer during processing. Each trace buffer entry describes current data capture status. If a severe error occurs, the Capture program prints the trace buffer before termination. The printing of the trace buffer supplements the Capture program error message.

Trace Output

When an error occurs, you can run the Capture program with the TRACE option. When you use this option, IBM Replication writes trace information logic flow to STDOUT.

Storage Dump

When Capture for VM terminates with a severe error, it saves critical diagnostic data in the CEEDUMP data set. This information is more detailed than the information in the Capture program trace buffer.

Operating Capture for VM

The replication administrator can use the commands in this section to operate Capture for VM.

This section explains how to perform the following Capture for VM tasks:

- Starting
- Stopping
- Suspending
- Resuming
- Reinitializing
- Pruning

Before You Start the Capture program

Before starting the Capture program, make sure you complete the following post-installation tasks:

- Define one or more replication sources and subscriptions as described in Chapter 8, “Working with Replication Sources” on page 93 and Chapter 9, “Working with Replication Targets” on page 103. This creates the following control tables that must be defined in the database where the Capture program is to run:
  - ASN.IBMSNAP_REGISTER
  - ASN.IBMSNAP_PRUNCNTL
  - ASN.IBMSNAP_CCPPARMS
  - ASN.IBMSNAP_TRACE
  - ASN.IBMSNAP_WARM_START
  - ASN.IBMSNAP_UOW
  - ASN.IBMSNAP_CRITSEC
- Ensure that the administrator has DBA privilege.
- Ensure that ASN.IBMSNAP_REGISTER has at least one entry in it by defining a replication source with the DATA CAPTURE CHANGES option.
- Enable the DATA CAPTURE CHANGES attribute for your source tables using either the CREATE TABLE or ALTER TABLE statement.

A Capture program ASNPARMS file is provided that contains default values that the Capture program uses. Changing this file modifies the defaults. If you need different values for a specific database, copy the file to the Capture program virtual machine's A-disk. The following default values are contained in the Capture program ASNPARMS:

- ENQ_NAME CAPTURE
- LANGUAGE ASNLS001

By default, the messages are issued in American English (ASNLS001). To issue messages in a different language, change the LANGUAGE parameter in the ASNPARMS file. Supported languages include:

- ASNLS001 - American English
- ASNLS002 - Uppercase English

- ASNLS003 - French
- ASNLS004 - German
- ASNLD001 - Japanese
- ASNLD002 - Chinese

Starting Capture for VM

After it starts, the Capture program runs continuously until you stop it or it detects an error.

To start the Capture program:

- Invoke the ASNCCP module from a VM user ID. Keywords must be separated by one or more blanks. The syntax is:

```
►►──ASNCCP──┬────────┬──┬─────────┬──┬─────────┬──┬─────────┬──────────►
            ├─WARM───┤  ├─PRUNE───┤  ├─TERM────┤  ├─TRACE───┤
            ├─COLD───┤  └─NOPRUNE─┘  └─NOTERM──┘  └─NOTRACE─┘
            └─WARMNS─┘

►──┬──────────────────────────────┬─────────────────────────────────►◄
   ├─NOUSERID─────────────────────┤
   └─USERID=──auth_name/password──┘
```

If conflicting invocation parameters are specified, Capture for VM uses the value of the last parameter specified. For example, if ASNCCP is started using the COLD TRACE NOTRACE parameter string, no informational event trace will be written (NOTRACE).

Table 18 on page 220 defines the optional parameters.

Table 18. ASNCCP Command Parameter Definitions for VM

Parameter	Definition
Src_server	Source server name positional parameter that must be the first parameter if entered. If not specified, the value from the DB2DBDFT environment variable will be used.
WARM (default)	The Capture program resumes processing where it ended in its previous run if warm start information is available. If the Capture program cannot warm start, it switches to a cold start. See “Warm and Cold Starts” on page 156 for more information.
WARMNS	The Capture program resumes processing where it ended in its previous run if warm start information is available. Otherwise, it issues a message and terminates. With WARMNS, the Capture program does not automatically switch to a cold start. The Capture program leaves the trace, UOW, change data, and warm start tables intact. In case of errors, the Capture program terminates instead of switching to a cold start as when WARM is specified. See “Warm and Cold Starts” on page 156 for more information.
COLD	The Capture program starts by deleting all rows in its change data, UOW, pruning control, and trace tables during initialization. See “Warm and Cold Starts” on page 156 for more information.
PRUNE (default)	The Capture program automatically prunes the change data and UOW tables.
NOPRUNE	Automatic pruning is disabled. The Capture program prunes the change data and UOW tables when you enter the PRUNE command. See the PRUNE command for more information.
NOTRACE (default)	No trace information is written.
TRACE	Writes debug trace messages to the standard output, *stdout*.
USERID=*auth_name/password*	Specifies that the Capture program should connect to the database as user ID *auth_name*. The correct password must be provided or an error is returned. The *auth_name* and password are both from 1 to 8 characters in length. For VM/ESA, if you do not specify this parameter, the Capture programconnects to the database as the user ID on which you issue ASNCCP.

Stopping the Capture program for VM

```
►►─STOP────────────────────────────────────────────────────►◄
```

Purpose

Use the STOP command to stop the Capture program gracefully and commit the log records that it processed up to that point.

STOP should be issued before:

- Removing an existing replication source

- Opening an existing replication source and modifying an existing replication source.
- Shutting down the database

Usage

If you stop the Capture program, it shuts itself down and issues an informational message. If it detects an error, the program shuts itself down after cleaning up the data in the affected tables (so that the data will not be used). Staging tables are pruned when it is appropriate. In the case of re-IPL or logoff, you must initiate a cold start because the warm start information could not be saved.

Capture for VM issues informational or error messages to the console.

Suspending Capture for VM

```
►►──SUSPEND────────────────────────────────────────────────────────►◄
```

Purpose

Use the SUSPEND command to suspend the Capture program until you issue the RESUME command.

Usage

You can use this command to suspend the Capture program to improve performance for operational transactions during peak periods without destroying the Capture for VM program run environment.

Resuming Capture for VM

```
►►──RESUME─────────────────────────────────────────────────────────►◄
```

Purpose

Use the RESUME command to resume the suspended Capture program.

Usage

You can use this command to resume the Capture program if you suspended it with the SUSPEND command.

Reinitializing Capture for VM

```
►►──REINIT─────────────────────────────────────────────────────────►◄
```

Purpose

Use the REINIT command to reinitialize the Capture program.

Usage

Use the REINIT command to begin to capture changes from new source tables if you add a new replication source with the Control Center while the Capture program is running. This command ensures that the Capture program recognizes the new replication sources in the change data control table. To allow the new replication source to be visible to the Capture program, you must issue a REINIT command. The REINIT command tells the Capture program to obtain newly added replication sources from the change data control table (ASN.IBMSNAP_REGISTER).

REINIT also rereads the tuning parameters table (ASN.IBMSNAP_CCPARMS) for any changes made to the tuning parameters.

Attention: Do not use REINIT to reinitialize the Capture program after canceling a replication source or dropping a replication source table while the Capture program is running. Instead, stop the Capture program and use warm start.

Pruning the Change Data and Unit-of-Work Tables

```
►►──PRUNE──────────────────────────────────────────────────────────►◄
```

Purpose

Use the PRUNE command to initiate the pruning of the change data (CD) and unit-of-work (UOW) tables, if you used the NOPRUNE invocation parameter on ASNCCP.

Usage

This command prunes tables once. During pruning, if you stop or suspend the Capture program, pruning does not resume after you enter the RESUME command. You must enter the PRUNE command again to resume pruning.

Providing the Current Log Sequence Number

```
►►──GETLSEQ────────────────────────────────────────────────────────►◄
```

Purpose

Use the GETLSEQ command to provide the timestamp and current log sequence number.

Usage

Use the GETLSEQ command to determine how far the Capture program has read the log.

Warm and Cold Starts

This section explains how the Capture program handles warm starts, how it switches to an automatic cold start, and when you might want to force a warm start.

Warm Start Process

When you start the Capture program with the WARM or WARMNS parameter, it searches for the warm start table, ASN.IBMSNAP_WARM_START, which was created at installation. This table contains information that enables the Capture program to quickly resynchronize to the time when it stopped. If this table is not available, the Capture program can resynchronize using either the common replication sources table, unit-of-work table, or change data tables.

The Capture program switches to a cold start if you did not specify WARMNS and the warm start log sequence number is not available in the DB2 3.1 active log, or if it is not available in the DB2 4.1 active or archived logs. (The Apply program performs a full refresh after a cold start for point-in-time tables; for change aggregate tables, gap messages are issued.)

Warm start information is saved in most cases. In extreme cases, warm start information might not be saved. For example, an operator might cancel the Capture program or stop DB2 with SQLEND QUICK in a VM environment. In this case, the Capture program uses the change data, unit-of-work, or change data control tables to resynchronize to the time it was stopped.

After a successful warm start, the old rows in the warm start table are deleted.

Automatic Cold Starts

Sometimes the Capture program automatically switches to cold start, even when you specify a warm start. (However, the switch to a cold start is not made when WARMNS is specified.) The switch is made when:

- The warm start log sequence lags behind the current log sequence by more than the LAG_TIME value as specified in the tuning parameters table, ASN.IBMSNAP_CCPPARMS.
- No warm start information is available, because it could not be saved.
- The warm start log sequence is not available on the DB2 active log. For example, this could happen if a COLDLOG was performed or a restore from an older archive was performed.
- The Capture program cannot keep up with the amount of log activity, and the log wraps.
- You invoke the Capture program for the first time.

In each of these cases, the Capture program issues an informational message and performs a cold start.

Forcing a Warm Start

You might want to prevent the Capture program from cold starting in some situations. For instance, the Capture program cold starts if DB2 goes down, and the NOTERM parameter was not specified. Forcing a warm start with the WARMNS parameter ensures that the control tables remain intact. You must correct the problem that caused the Capture program to terminate. If you do not correct the problem, the Capture program continues to terminate every time you start it.

Troubleshooting: Problems Using the Capture program

This section describes commonly known issues with the Capture program. Before running Capture for VM, ensure that you have all the PTFs for your level of DB2. Specific Capture program problems are described below.

Problem

Capture for VM does not start.

Ensure that:

- Access has been given for the database log and directory minidisks. Note that the Capture program issues internal links to these minidisks.
- Access has been given to the C Run Time Library.
- *IDENT authorization has been given to the Capture program virtual machine.
- The ASNLMAIN package file has been loaded to the database.

Problem

The Capture program is not capturing updates.

Any of the following could prevent Capture for VM from capturing updates:

- DATA CAPTURE CHANGES was not specified on the base tables to be captured.
- The proper order for starting the Capture and Apply programs was not used:
  1. Replication sources and subscriptions must be defined before starting Capture for VM.
  2. Start Capture for VM and look for message number ASN0100I (initialization completed) on the virtual machine console or in the ASN.IBMSNAP_TRACE table.
  3. Start the Apply program.

Check the ASN.IBMSNAP_TRACE table for possible error messages.

Problem

I'm not sure if the Capture program is running successfully.

After you start the Capture and Apply programs, the Apply program performs a full refresh to populate the target tables. Then the Capture program writes message ASN0104I to the ASN.IBMSNAP_TRACE table, providing information related to table owner name, table name, and starting log sequence number value. This provides a point from which the Capture program starts to capture updates.

Subsequent captured updates are placed in change data tables and eventually applied to target tables and pruned from the change data table. After the Capture program runs for some time, you should see rows in the change data table if changes are made to the source. Periodically, check the ASN.IBMSNAP_TRACE table to see the progress made by the Capture program. If it encounters errors, it sends them to the console and also logs them in the trace table. Similarly, the Apply program logs its information in the *userid*.IBMSNAP_TRAIL table.

Problem

Capture for VM issued message ASN0000E instead of the proper message number.

If you receive message ASN0000E, it implies that a generic message was issued instead of a proper message. This happens if either the default message file, ASNLS001 MSG, or the specified message file in CAPTURE ASNPARMS was not found. See the Capture for VM program directory for information on installing the message file.

Problem

Capture for VM terminated unexpectedly.

The Capture for VM program terminates either because of a severe error, or when you issue the STOP command. The Capture program terminates with a return code that indicates successful or unsuccessful completion. Return codes are:

0	STOP command issued
8	Error during initialization
12	Any other severe error

Part 5. Advanced Topics

This part contains advanced topics such as mobile replication, advanced techniques, and problem determination.

Chapter 16. Mobile Replication

This chapter provides a brief overview of mobile replication, steps to plan and implement mobile replication, a description of mobile replication, and an explanation of how to run the mobile replication enabler program from the command line or from the mobile graphical interface.

An Overview of Mobile Replication

Mobile replication provides you with the flexibility that you need to efficiently manage your data on the go. Instead of trying to guess when you will be able to connect and synchronize your laptop computer with the home office, the mobile replication enabler allows you to transfer data on demand. The mobile replication enabler, also referred to as the ASNCOPY program, allows you to transfer data on demand because it controls the execution of the Capture and Apply programs.

You are not burdened with cumbersome manual procedures and high telecommunications costs. Instead the new mobile graphical interface, also referred to as the ASNMOBIL program, provides you with an interface to quickly and easily select and send data, as well as receive it. Typing ASNMOBIL on the command line will start the ASNMOBIL program. The mobile graphical interface can then be used to invoke the ASNCOPY program. Mobile replication also minimizes the frequency and duration of communication line connections, thus reducing telecommunications costs.

The mobile replication enabler and its graphical interface invocation program, ASNMOBIL, is supported on the following platforms:

- OS/2
- Windows NT
- Windows 95

The mobile replication enabler can replicate data to and from source servers on the following platforms:

- OS/2
- Windows NT
- AIX
- Sun Solaris
- HP
- MVS
- VM
- VSE

Highlights

Mobile replication highlights include the following features:

- Automatic connect and disconnect of your mobile client via user exit programs
- Automatic start and stop of the Capture and Apply programs
- Copy on demand

- The ability to select replication subscriptions
- Telecommunication cost optimization

How Mobile Replication Works

The mobile replication components are initiated with the mobile replication enabler, ASNCOPY. You can invoke the mobile replication enabler program from the Mobile Replication Enabler window, from the command line, or from an application program.

The mobile replication enabler replicates selected replication subscriptions as soon as possible and ignores relative timing or event timing options. It uses the same control tables that are used in non-mobile replication.

You can seamlessly make the switch between mobile and non-mobile modes, without needing to redefine subscriptions.

Mobile Replication Restrictions

Mobile replication has the following restrictions:

- The mobile replication enabler can support only one control server at a time. This control server is the default DB2 database located on your mobile client.
- The mobile replication enabler does not provide a dial-up or disconnect program. You can specify your own user exit programs via the ASNDIAL and ASNHANGUP environment variables.
- Security for mobile replication must be provided by the underlying software products such as the DB2 Universal Database or the dial-up program for user authentication and DBMS access authorization.

Planning Mobile Replication

This section explains how to get your laptop up and running for mobile replication. This section discusses:

- Software and hardware requirements
- Communication program requirements
- Configuring the mobile client

Software and Hardware Requirements

In addition to the non-mobile IBM Replication hardware and software, mobile replication also requires communication hardware and software to transfer data. This section lists probable hardware and software needs. This list is not exhaustive, because it would be impossible to list all the hardware and software requirements for the unlimited number of possible configurations.

- Communication hardware
  - Adaptors
  - Modems
  - Phone lines

- Communication protocols
  - Netbios
  - TCP/IP
  - SNA
- Communication software
  - IBM LAN Distance

Communication Program Requirements

To ease the burden of manually dialing up and disconnecting your communications lines, your database administrators can specify a program to dial and disconnect your communications lines for you. Although IBM Mobile Replication does not provide any connecting or disconnecting communication programs, it does provide a means to automate your user-developed connection/disconnection communication programs via two environment variables: ASNDIAL and ASNHANGUP.

ASNDIAL Specifies the dial-up user exit program. When specified, the mobile replication component calls the dial-up program every time a physical connection is needed. IBM Mobile Replication does not pass any parameters and does not expect any return code from this exit program.

ASNHANGUP Specifies the disconnect user exit program. When specified, the mobile replication component calls the disconnect program as soon as the line is no longer needed. The only exception is when the disconnect program has been disabled by the hold-line (-H) ASNCOPY invocation option. You can disable the automatic disconnect function for a variety of reasons. For example, you might want to issue DB2 commands against the source server after ASNCOPY is finished, or you might need a repeated copy to successfully copy a large answer set. IBM Mobile Replication does not pass any parameter and does not expect any return code from this exit program.

Specifying ASNDIAL and ASNHANGUP Environment Variables in OS/2

The following section describes how to specify ASNDIAL and ASNHANGUP environment variables in OS/2.

To set the ASNDIAL and ASNHANGUP environment variables:

1. Declare the environment variables in your config.sys file. For example:

   - `SET ASNDIAL =` *C:\sqllib\bin\mydial.exe*
   - `SET ASNHANGUP =` *C:\sqllib\bin\myhangup.exe*

   Where:

   mydial.exe
   The program you use to connect your mobile client to the source server.

myhangup.exe
The program you use to disconnect your mobile client from the source server.

2. Reboot your system to have these settings take effect.

Specifying ASNDIAL and ASNHANGUP Environment Variables in Windows NT

The following section describes how to specify ASNDIAL and ASNHANGUP environment variables in Windows NT.

To set the ASNDIAL and ASNHANGUP environment variables:

1. From the **Control Panel** window, double click on the **System** icon. The System Properties notebook opens.
2. Select the **Environment** tab.
3. Set the ASNDIAL variable:
   a. In the **Value** field, type the path for your user-defined connect program. For example:

      ```
      C:\sqllib\bin\mydial.exe
      ```

   b. Click on **Set**.
4. Set the ASNHANGUP variable:
   a. In the **Value** field, type the path for your user-defined disconnect program. For example:

      ```
      C:\sqllib\bin\myhangup.exe
      ```

   b. Click on **Set**.
5. Click on **OK**. The environment variables are set.

Specifying ASNDIAL and ASNHANGUP Environment Variables in Windows 95

The following section describes how to specify ASNDIAL and ASNHANGUP environment variables in Windows NT.

To set the ASNDIAL and ASNHANGUP environment variables:

1. From the **Control Panel** window, double click on the **System** icon. The System Properties notebook opens.
2. Select the **Environment** tab.
3. Set the ASNDIAL variable:
   a. In the **Value** field, type the path for your user-defined connect program. For example:

      ```
      C:\sqllib\bin\mydial.exe
      ```

   b. Click on **Set**.
4. Set the ASNHANGUP variable:

a. In the **Value** field, type the path for your user-defined disconnect program. For example:

```
C:\sqllib\bin\myhangup.exe
```

b. Click on **Set**.

5. Click on **OK**. The environment variables are set.

Configuring the Mobile Client

You configure the mobile client just as you would configure the target server in non-mobile replication for the respected platforms. See Chapter 3, “Getting Started” on page 39 for more information on how to configure target servers.

Setting Up the Mobile Client

Define registrations and subscriptions in the same way that you define registrations and subscriptions in the non-mobile replication process. The unique situation created in the mobile environment is that the mobile client may not always be connected to the Control Center. When the control server is disconnected from the Control Center, you must find a way to get registration and subscription information to the mobile client.

There are two possible ways to distribute the registration and replication subscription information needed to run a mobile replication scenario. Each of the possibilities has its own restrictions.

- Install the DB2 Control Center on the laptop. While the mobile client is connected to the source server, define registrations and subscriptions.
- Define registrations and subscriptions using the Control Center on another server, save and customize the SQL files, download the files via diskette or FTP, and run the SQL files on the mobile client.
  1. Define registrations and replication subscriptions at the Control Center by connecting to the source server.
  2. Save and edit the generated SQL files. Customize the values for the CONTOL_SERVER, CNTL_ALIAS, SOURCE SERVER, and TARGET SERVER columns of ASN.IBMSNAP_SUBS_SET table to match those on the mobile client.
  3. Transfer the customized SQL files and DPCNTL.UDB to the mobile client via a diskette or by downloading the information.
  4. Run the DPCNTL.UDB files on the mobile client.
  5. Run the customized SQL files on the mobile client.

Defining the Control Server for Your Mobile Client

The mobile replication enabler uses the default database on the mobile client as its control server. You must specify the default database and instance name by setting the environment variables as follows:

- SET DB2DBDFT = *control_server_name*
- SET DB2INSTANCE = *database_instance_name*

If your database environment does not match the default database environment, make the necessary changes so that the default database and instance is set properly. For more information about how to define replication sources and subscriptions see Chapter 8, "Working with Replication Sources."

Mobile Replication Processing Cycle

When the mobile replication enabler is invoked, the Capture and Apply programs replicate data as defined in the replication subscription in the following manner:

1. ASNCOPY calls the Capture program. The Capture program stops when it has captured all the changes up to the time when it was called.
2. ASNCOPY sets the value of the ACTIVATE column in the ASN.IBMSNAP_SUBS_SET table to 2, which requests an immediate copy.
3. ASNCOPY calls the dial-up user exit program, if specified in the ASNDIAL environment variable.
4. ASNCOPY calls the Apply program.
5. The Apply program replicates all of the data sets that are marked for immediate replication. Replication is repeated if the sizes of the answer sets are regulated by MAX_SYNCH_MINUTES.
6. The Apply program calls the disconnect user exit program, if specified in the ASNHANGUP environment variable, at the earliest opportunity to minimized the phone cost.
7. ASNCOPY displays replication statistics from the information selected from the ASN.IBMSNAP_APPLYTRAIL table.
8. ASNCOPY then stops.

Starting the Mobile Replication Enabler Using the ASNCOPY Command

ASNCOPY, the mobile replication enabler, is the program that controls the execution of the Capture and Apply programs. You can specify parameters to run the mobile replication enabler when you invoke ASNCOPY from the command line. You can also specify mobile replication enabler parameters from the mobile graphical interface. To execute the mobile graphical interface, type ASNMOBIL from the command line. For more information about the mobile graphical interface see “Starting the Mobile Replication Enabler Using the Mobile Graphical Interface” on page 235

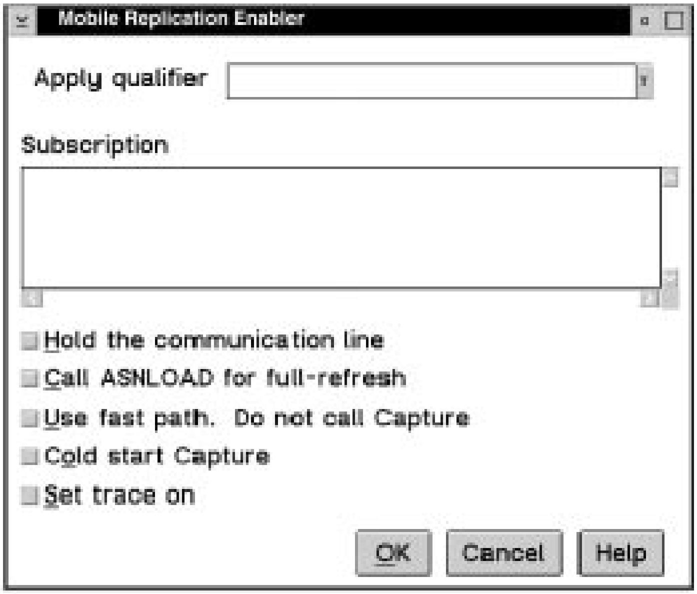
Table 19 on page 235 lists the parameters that you can specify for the ASNCOPY command. These parameters are not case sensitive.

Table 19. ASNCOPY Command Parameters

Parameter	Explanation
-T	Activates the trace option. You can set the trace option on from the Mobile GUI. When the trace is set on, information about all database activity will be written to a file called ASNCOPY.TRC. The Apply program trace is written in a file called <apply_qual>.TRC, where <apply_qual> is the Apply qualifier value, for the OS/2 platform and in a file called APPLYYYYYMMDDHHMMSS.TRC for the Windows NT and 95 platforms. The Capture program trace is written in the trace table called ASN.IBMSNAP_TRACE.
-H	Holds the communication line connection even after all answer sets are fetched. When this parameter is not specified, the Apply program disconnects the link between the target and source servers by calling the disconnect user exit program as soon as all of the answer sets are fetched. This is done to minimize the line connection time. If you do not specify this parameter, you might not be able to repeat a copy, which might be needed if the target server log file overflows. This problem is not detected until after the link is disconnected. Retry the failed copy with this parameter specified. If overflow is detected before the link is disconnected repeat copies are attempted regardless of whether this parameter is specified.
-L	Specifies that the Apply program call the ASNLOAD user exit program for a full refresh. In the ASNLOAD user exit program ,you can invoke either the IBM provided EXPORT/IMPORT or EXPORT/LOAD utilities, or any other programs for a faster full refresh. If referential constraints are defined between member tables of a replication subscription, you must use this option to bypass RI constraint checking while the tables go through a full refresh.
-F	Specifies the fast path. This parameter tells ASNCOPY not to call the Capture program. If this parameter is not specified, the Capture program is always called regardless of the copy direction. Use this option only if either no changes have been made to the local tables since the last Capture program run or the replication subscription to which all the changed tables belong is not to be copied. Otherwise, the changes will be lost.
-C	Specifies that ASNCOPY starts the Capture program with a cold start. If a cold start occurs, then all target tables are refreshed from the source tables and all changes made locally that were not applied to the source server are lost. If this parameter is not specified, ASNCOPY uses the WARMNS start option. The -C parameter is ignored if the fast path (-F) parameter is specified or if the Capture program is not installed.
-Q *apply_qualifier*	You must specify this parameter when the Apply qualifier is not the same as the user id.
-S *set_name_list*	Specifies a list of replication subscription names that you want to copy.
-U *set_name_list*	Specifies the copy up (WHOS_ON_FIRST = F) legs of replica sets.
-D *set_name_list*	Specifies the copy down (WHOS_ON_FIRST = S) legs of replica sets.

Starting the Mobile Replication Enabler Using the Mobile Graphical Interface

The mobile graphical interface, Mobile Replication Enabler window, can be used to start the mobile replication enabler, ASNCOPY. To initiate the mobile graphical interface, type ASNMOBIL from the command line. The Mobile Replication Enabler window opens, see Figure 32 on page 236.



Mobile Replication Enabler

Apply qualifier

Subscription

□ Hold the communication line

□ Call ASNLOAD for full-refresh

□ Use fast path. Do not call Capture

□ Cold start Capture

□ Set trace on

Figure 32. The Mobile Replication Enabler Window. You can set ASNCOPY parameters.

From the Mobile Replication Enabler window, you can:

- Set the trace on
- Hold the communications line
- Call ASNLOAD for full refresh copying
- Use the fast path
- Specify the cold start option for the Capture program

These options are explained in more detail in Figure 32.

The Mobile Replication Enabler window also allow you to select replication subscriptions and an Apply qualifier.

Selecting Replication Subscriptions

The mobile graphical interface presents a list of active replication subscriptions from which you can select the replication subscriptions you wish to run. A replication subscription contains the specification of the source and target tables, as well as the control information that governs a refresh or update.

The mobile replication enabler, ASNCOPY, will run with the replication subscription names you selected. A complete list of replication subscriptions can be found in the ASN.IBMSNAP_SUBS_SET table on the control server.

Selecting an Apply Qualifier

An Apply qualifier is a unique name in the network to distinguish one instance of the Apply program from another. To populate a replication subscription list with a particular replication subscription, simply select the Apply qualifier that represents the Apply instance running on your client. ASNCOPY will now only fetch data according to the replication subscription associated with the Apply qualifier selected.

Chapter 17. Advanced User Techniques

This chapter describes two techniques for the advanced IBM Replication user:

- Defining Replication Sources and Targets for Update-Anywhere Replication
- Stopping the Capture Program with a Trigger

Defining Replication Sources and Replication Subscriptions for Update Anywhere

The following steps describe how to set up the sources and targets for update-anywhere replication. They do not list all the possible steps for defining the source and subscription; just those of particular interest when administering an update-anywhere scenario.

To define a replication source and subscription for update anywhere:

1. Define a custom replication source, using the following selections. See “Defining Replication Sources” on page 93 to learn how to define a replication source.
    a. Ensure that the **Define as Source** check boxes is selected for every column.
    b. Ensure that the **Capture before image** check boxes is selected for every column.
    c. Click on the **Table will be used for update anywhere** check box.
    d. Select a conflict detection level.
2. Define a replication subscription set using the following selections. See “Defining Replication Subscriptions” on page 103 for information about the steps for this task.
    a. Select the replication sources to be in the subscription set. Include all sources affected by the replica tables being updated.
    b. From the Subscription Definition window, select a target table to be defined as a replica, an updateable target table.
    c. Click on **Advanced** to open the Advanced Subscription notebook. The following sections are required on the Advanced Subscription notebook:
        1) From the Target Table page, click on **Target table is replica**.
        2) From the Target Columns page:
            a) Ensure that the **Subscribe** check boxes are selected for every column. *Do not* create new columns for the replica table.
            b) Specify a primary key for the replica table by clicking on the **Primary Key** check boxes next to the key column names.

                Note: Make the primary key the same as the source table primary key to prevent conflicts.

        3) If you want the replica to be a subset of the source table, from the Rows page, type a row predicate in the **WHERE** field.

        4) Click on **OK** to close the Advanced Subscription notebook.

    d. Repeat the Advanced Subscription notebook steps for each target table.

    e. Click on **Timing** to open the Subscription Timing notebook.

        1) On the Source to Target page, fill in the subscription set timing information for copying the source tables changed data to the target tables.

        2) On the Replica to Target page, fill in the subscription set timing information for copying the replica tables changed data to the source tables.

        3) Click on **OK** to close the notebook.

    f. If you want to define SQL or CALL procedures to run before or after the subscription set is processed, click on **SQL** and define the processing statements.

Stopping the Capture Program with a Trigger

In some situations, you might want to trigger the stop of the Capture program. For example, you might want to stop it at a specific time, or after all transactions for a particular application have been committed. The simplest way to set up this trigger is to create a table (or use an existing table), define it as a replication source, and identify a particular column whose update, when captured, triggers shutdown.

To stop the Capture program with a trigger:

Use the following example as a guideline.

1. Create a table with a column named STOPCAP or another unique identifier.
2. Define the table as a replication source.
3. Create an application that periodically reads the change data table associated with this replication source table to search for STOPCAP.
4. Schedule the application to start before your Apply programs.

When your application encounters STOPCAP in the change data table, it should issue the command to stop the Capture program.

Chapter 18. Problem Determination Facilities

This chapter provides basic information for using IBM Replication problem determination facilities, such as message explanation, trace and log records, and control table information. It also provides a high-level example of how to use the information to trace and the error and a list of question often asked by IBM service personnel that can help you to prepare to work with IBM Software Support. Use the information in this chapter to gather information about usage and operational errors so that you can work with IBM Software Support to resolve software problems.

IBM Replication provides the following facilities for problem determination:

- Error messages and SQL states for the Control Center, the Capture program, and the Apply program
- The Apply program trail table, log file, and trace file
- The Capture program trace table, log file, and trace file
- IBM Replication control tables and files

When using the problem determination facilities to test or debug your replication scenarios, we recommend that you:

- Attempt any new replications in a test environment.
- Stop other replication activity while gathering information about a problem in order reduce the volume of data to sift through.

See “Problem Source Identification Questions” on page 246 for a list of questions that can help you to research the error condition.

Replication Diagnosis Resources

The following section describes resources for determining how to diagnose replication errors.

Errors Encountered during Replication Administration

The Control Center can encounter errors either when it is gathering information from source servers, target servers, or control servers to create the SQL statements for administration or when it is actually running the SQL to set up the replication sources and subscriptions. The primary indicators are SQL messages and SQL states that accompany the error. The SQL messages and states are issued in error message windows in the Control Center.

The IBM Replication tools are primarily relational database applications. The replication control tables are created using DDL issued by the Control Center, and data is replicated primarily by SQL SELECT, INSERT, UPDATE, and DELETE statements issued by the Capture and Apply programs. When an error occurs during replication administration tasks, the error messages are normally relational database error messages, such as SQL messages and SQL states. See the DB2 message reference for your platform for more information about DB2 error messages and SQL states.

Errors Encountered While Running the Capture and Apply Programs

The Capture and Apply programs can encounter a problem while capturing and replicating changed data, even though the SQL that the Control Center generated for defining replication sources and subscriptions ran without error. You can determine the cause of the errors with information in the following locations:

- SQL messages and SQL states found in the Apply trail control table
- The Apply program trace file
- The Capture program trace control table

The Capture and Apply programs issue their own messages. The messages for the Capture and Apply programs begin with the letters ASN. Explanations and user response information are provided in this book in Chapter 21, “IBM Replication Messages” on page 303, in the *DB2 Universal Database Messages Reference*, and in DB2 Universal Database online help.

The messages for the Capture and Apply programs are issued or recorded in the following locations:

- At the command line processor window or console from which the Capture and Apply programs are started,
- In the Apply trail control table (ASN.IBMSNAP_APPLYTRAIL) and the Capture program trace table (ASN.IBMSNAP_TRACE)
- In the trace files for the Capture and Apply programs
- In the log files for the Capture and Apply programs

The Apply Program Problem Determination Facilities

SQL and Apply program error messages can be found in the ASN.IBMSNAP_APPLYTRAIL table and the Apply program trace file. There is one trace file at the control server associated with the Apply program. The Apply program log file tracks the activities of the Apply program and can be a useful diagnosis tool.

The Apply Trail Control Table (ASN.IBMSNAP_APPLYTRAIL)

There is an Apply trail control table (ASN.IBMSNAP_APPLYTRAIL) located at each control server with the subscription control tables, such as ASN.IBMSNAP_SUBS_SET. The Apply program inserts a new row in the Apply trail control table every time it attempts to replicate a subscription. There is a row for all successful and unsuccessful subscription cycles of each replication subscription. For a description of the ASN.IBMSNAP_APPLYTRAIL, see Chapter 20, “Table Structures” on page 265.

The Apply trail control table records one SQL code and one SQL state for a replication subscription that does not get replicated successfully. Additional SQL codes and states associated with the problem can be found in the Apply program trace file.

You can query the Apply trail control table, ASN.IBMSNAP_APPLYTRAIL for information about successful and unsuccessful replications. Some key fields in the table that have problem indicators are:

STATUS	Contains -1 to indicate a failed execution.
SQLSTATE	Contains the error SQLSTATE for a failed execution.
SQLCODE	Contains the error SQLCODE for a failed execution.
SQLERRM	Contains the text of the SQL error message.
APPERRM	Contains the text of the Apply program error message.

Within the error message text, determine which database the Apply program was connected to when the error occurred; for example, did the error occur while the Apply program was connected to the source server or the copy server?

The Apply trail control table has fields that identify the source and target databases and tables so that you can locate the Apply trail control rows that are causing replication errors. To reduce the number of rows that you examine, you can:

- Delete all rows from the ASN.IBMSNAP_APPLYTRAIL table, to clean out rows from past replications, before starting the Apply program.
- Temporarily disable replications that are successful in order to capture rows in ASN.IBMSNAP_APPLYTRAIL only for replications that have problems before starting the Apply program.

An example query for the ASN.IBMSNAP_APPLYTRAIL is:

```
SELECT * FROM ASN.IBMSNAP_APPLYTRAIL
SELECT TARGET_TABLE, STATUS, SQLSTATE, SQLCODE, SQLERRM, APPERRM
FROM ASN.IBMSNAP_APPLYTRAIL
```

Apply Program Trace File

The Apply program creates a trace file when the Apply program trace invocation parameter is used. See the Capture and Apply chapter for your platform in this book for the Apply program invocation command.

If you specify a trace option, specify the name of a trace output file and, for workstation systems, precede the output file name with a pipe symbol (>). For example, to start Apply for Windows NT with trace, issue the following command from the command line processor window:

```
\APPLY>asnapply myapply mydbnt2 trcflow > apply.trc
```

Where:

myapply is the Apply qualifer.

mydbnt2 is the control server where the Apply program finds the control tables.

trcflow indicates that the Apply program traces all error and flow information.

apply.trc is the file to which the output is directed.

The trace file is located in the same directory from which the Apply program is started.

After the Apply program is stopped, you can view the trace file with any editor. You can also transmit the file to other systems, such as by FTP, or print it.

You have two trace options:

TRCFLOW Provides very detailed information and is oriented toward helping IBM Service diagnose errors. When using TRCFLOW, we suggest isolating the subscription error, such as by running it in a test environment or disabling other error-free replication subscriptions to reduce the volume of information.

TRCERR Provides less detail and is a better choice when you are new to the replication tools.

Within the trace, particularly with the TRCFLOW option, entries are made into the trace file for the Apply program's activities. The following are examples of the recorded information:

- Connecting to the control server to obtain information on replication subscriptions to be processed
- Connecting to source servers to fetch rows to be replicated from the CD table to the target table
- Connecting to the target servers to insert, update, and delete rows into and from the target tables

The Apply program inserts error messages and indicators in the trace file at points when it encounters an error.

The Apply Program Log File

The Apply program also has a log file containing messages with a summary of the Apply program's activities. The log file is in the same directory from which the Apply program is started and where the *.SPL file and any trace files are located.

The name of the Apply program log file is the Apply qualifier associated with the control server with the extension of *.LOG. For example, for an Apply program operating with a Apply qualifier of MYAPPLY, the log file name is MYAPPLY.LOG.

Because the Apply program log file information is high level, it typically directs you to the ASN.IBMSNAP_APPLYTRAIL table for more detailed information.

Capture Program Problem Determination Facilities

The Capture program has a trace table, a log file, and trace file generated when the trace invocation parameter is used to start the Capture program.

Capture Program Trace Control Table (ASN.IBMSNAP_TRACE)

The trace control table is located in source server databases where the Capture program maintains change tables for replication sources. The trace control table, ASN.IBMSNAP_TRACE, contains basic information about the activities of the Capture program instance. For a description of the ASN.IBMSNAP_TRACE, see Chapter 20, "Table Structures" on page 265.

You can query the ASN.IBMSNAP_TRACE table with normal SQL (such as SELECT) or query tools. For example:

```
SELECT * FROM ASN.IBMSNAP_TRACE
```

Capture Program Trace File

The Capture program can be started with the trace invocation parameter. This parameter specifies that problem determination records be recorded in standard output to the screen or to a file, which contains the Capture program internal logic flows. When starting the Capture program, on workstation systems, precede the output file name with a pipe symbol (>).

To start the Capture program with trace on a workstation system:

```
\CAPTURE>asnccp mysrcdb x x trace > cap.trc
```

Where:

mysrcedb is the source server database for this Capture program instance.

x x indicates that no parameters are provided for the type of start (WARM, WARMNS, COLD) and pruning (PRUNE or NOPRUNE) so the defaults are assumed.

trace indicates that the Capture program traces all error and flow information.

cap.trc is the file to which the output is directed.

The trace file is located in the same directory from which the Capture program is started.

Capture Program Log

The Capture program creates a log file, named by the source database that is specified when the Capture Program is started; the file extension is *.CCP. If the Capture program is started with asnccp mysrcdb, the log file is named mssrcdb.ccp. The Capture Program log file is located in the same directory from which the Capture program is started.

Problem Determination Scenario

The following steps are a high-level description of how you might trace a replication error, using the facilities discussed in this chapter.

In this scenario, you used the Control Center to define replication sources and subscriptions. The SQL for your replication requests completed satisfactorily, but the Apply program can't execute the replication successfully. To determine the error, you could:

1. Examine any error messages returned directly to the terminal for the Apply program job or process.
2. Examine the Apply trail control table (ASN.IBMSNAP_APPLYTRAIL) for any indicators of the problem.
3. Examine the Capture program trace table (ASN.IBMSNAP_TRACE) for indicators from the Capture program's activity.
4. Examine the log files for the Capture and Apply programs for indicators from the activities of the Capture and Apply programs.
5. Rerun the Capture and Apply programs with the trace option and examine the trace file for indicators of the problem.

Problem Source Identification Questions

If you call IBM Software Support Services, you will be asked the following types of questions by Level 2 Service Support. You can save time and perhaps diagnose the error yourself by researching the answers to these questions.

1. What was occurring at the time of the problem?
2. What has changed recently in the environment?
3. Describe the environment.
4. What is the CSD level of DB2 Universal Database V5 where the Control Center is installed?
5. On what platform is the Capture program running?
6. At what maintenance level is the Capture program?
7. What is the maintenance level of DB2 where the Capture program is running?
8. On what platform is the Apply program running?
9. At what maintenance level is the Apply program?
10. On what release of DB2 does the Apply program run?
11. What is the maintenance level of DB2 where the Apply program is running?
12. Is this a mobile user?
13. What are the ASN messages that are issued?
14. Are there other messages either in SYSLOG or on the screen?
15. What is the complete message text for all messages? Be sure to note all message numbers, database names, table names, user IDs, and file names that appear in messages.
16. Where does the failure occur?

a. The Control Center

   - Is the problem with a replication source or subscription?
   - What messages appear?
   - Can the user successfully connect to the source or target database from a command line or command prompt window?

b. The Apply program

   - Is the Apply program running?
   - If not, what occurs when the Apply program starts?
   - What messages appear?
   - Is there error information in the ASN.IBMSNAP_APPLYTRAIL table?
   - Is there error information in the Apply program log file?
   - Are data changes being successfully replicated to the target table?
   - Do all tables in a replication subscription have the same problem?
   - What table types (for example, user copy, point-in-time, CCD) are involved in the failure?
   - Did you run the Apply program with trace?
   - Are CALL procedures being used?
   - Is a CCD being used?

c. The Capture program

   - Is the Capture program running?
   - If the Capture program is not running, what happens when a warm start of the Capture program is tried?
   - Is there error information in the ASN.IBMSNAP_TRACE table?
   - Is there error information in the Capture program log file?
   - What is the DB2 configuration?
   - Are data changes being successfully inserted into the CD tables?
   - Does the user ID running the Capture program have sufficient privileges to run the Capture program?

Part 6. Reference Information

This part provides reference information for the replication tools. It describes the control and target table structures, how to migrate from the previous release, and provides all the of the messages issued by the Capture and Apply programs.

Chapter 19. Migrating from DataPropagator Relational Version 1 to IBM Replication Version 5

This chapter describes the steps involved in migration, the restrictions and assumptions, and how to use the migration program to migrate from the prior version of IBM Replication known as DataPropagator Relational.

If you are a Version 1 customer, you will be able to migrate your replication definitions by running the migration program. However, if you used certain advanced techniques, you might have definitions that are no longer valid for IBM Replication Version 5, thereby requiring manual intervention. For these situations, the migration program enables you to analyze and then migrate your unique replication definitions, if desired.

Migration Process Overview

From the perspective of the migration program, the migration process consists of the following steps:

- Collection
- Analysis
- Migration

This section provides a brief overview of each step. Detailed information about the steps and command syntax are provided later in this chapter.

Collection

The collection step consists of gathering the existing Version 1 replication definitions into a DB2 migration database. This migration database contains several global control tables, two of which store registration and subscription information from all specified Version 1 servers.

You can repeat the collection step, each time generating a new version of the registration and subscription details for analysis.

Analysis

The analysis step consists of analyzing the contents of the global control tables in the migration database to detect potential migration exceptions and make necessary adjustments.

The migration analysis step does the following:

- Updates the migration database with modified information that instructs the migration program.
- Writes a migration analysis report that explains the automated modifications and recommendations for migrating the Version 1 system.

After reading the migration report, you can decide whether the automated migration choices are appropriate. If not, you can override the program's results by using standard SQL to update the data in the global control tables. Then you can rerun the analysis step to validate the current migration instructions.

Migration

The migration step consists of reading the replication definitions from the migration database and Version 1 control tables, and migrating to Version 5 control table formats. The MIGRATE action migrates one source server, one target server, or one of each (source and target) at a time. Source server and target server are Version 5 terms that replace the DataPropagator Relational Version 1 terms of source server and target server. The Version 5 terminology is used throughout this chapter.

You can back out of Version 5 migration and return to Version 1. If you choose this action, the migration program reads the original Version 1 definitions from the original Version 1 tables and the log sequence numbers from the current Version 5 control tables.

After successful migration, you can drop unused Version 1 tables and views to restore disk space if needed by running MIGRATE with the CLEANUP action.

Migration Requirements

Consider the following points before migrating.

- Migration requires DBADM authority for the databases that contain the Version 1 and Version 5 control tables, and the migration global tables.
- Only one Capture instance per DB2 subsystem/database can exist at a time.
- You must migrate the Capture program to Version 5 before migrating any associated Apply program processes.
- To use the migration program for Windows 95 or Windows NT, you must create a password file. See “Providing End-User Authentication at the Source Server” on page 147 for more information.
- After migrating, you can use the Version 5 Control Center to create new Version 5 replication sources and subscriptions, but you cannot use the DataPropagator Relational Control Point for OS/2 to create Version 1 subscriptions.

Migration Precautions

Only transaction consistent is supported; convergent consistent is not supported. Consequently, when you migrate Version 1 registrations with CD_CONSISTENT=C, they may lose some compensation records. The subscriptions will continue to run without errors, however.

It is recommended that you not use the Version 5 Control Center when Version 5 Capture and Version 1 Apply coexist. However, if you need new replication sources, you can use the Version 5 Control Center to define Version 5 sources and subscriptions.

Installation of the Capture and Apply Programs on DB2 Version 2

You can independently install the Version 5 Capture and Apply programs without having to migrate your database to Version 5 of DB2. To do this, read the README file in the REPL directory of the DB2 Universal Database Version 5 CD for instructions.

Before You Begin Migration

Before you proceed with the migration steps in the following sections, be aware of the following points.

- The migration program can execute at a Windows NT, Windows 95, or OS/2 control point, which must be able to connect to every source, target, and control server.
- The initial value for the Version 5 APPLY_QUAL is the Version 1 Apply ID. This Apply ID will be the subscriber's ID supplied with the migration program.

  The term *Apply qualifier* (APPLY_QUAL) is a new concept in Version 5 subscription as well as a column in many subscription control tables. The value in APPLY_QUAL identifies the Apply instance that runs a given subscription. The PREPARE action uses given defaults to convert the Version 1 subscriber ID to a Version 5 APPLY_QUAL.
- The initial value for the Version 5 SET_NAME is the Version 1 COPY_TABLE.

  The term *replication subscription* (SET_NAME) has a new concept in Version 5. It groups many individual subscriptions to source tables on the same source server, where updates to all copies in the group are committed in a single target server transaction, and each copy reflects all source server updates up to a common source server synchronization point.
- The ASN.IBMSNAP_CCPPARMS, ASN.IBMSNAP_CRITSEC, and ASN.IBMSNAP_UOW tables remain in their altered format if fallback occurs.
- You must use the migration report to determine the order of registration and subscription migration.

Invoking the Migration Program and Actions

You can invoke the migration program by entering the following command followed by an action and two or more required parameters. If you enter the command without entering an action, syntax help is provided.

```
ASNMIG
action parameters
```

Where *action* is one of the following choices:

BUILDDB Required action that gathers Version 1 replication definitions and saves them in an ASCII file named BUILDDB.SQL (default name). You must execute this file using the DB2 -TF command to create the migration global control tables and insert rows during the BUILDDB run.

PREPARE Required action that either transforms incompatible Version 1 column values into compatible Version 5 values, or informs you about non translatable Version 1 values.

MIGRATE Required action that migrates from Version 1 to Version 5.

FALLBACK Optional action that reverts from Version 5 to Version 1.

CLEANUP Optional action that drops unused Version 1 tables and views after successful migration.

The following sections explain how to use each action and its associated parameters.

Collecting Data with BUILDDB

Enter the following command to collect Version 1 registration and subscription information from the servers in your replication environment:

```
ASNMIG BUILDDB
mig_db serv_list
```

Where:

mig_db Name of the migration database where you want to store the global control tables. The database must already exist before you can enter a value for this parameter. This is a required parameter.

serv_list List of databases from which you want to gather Version 1 replication information. At least one database alias name is required. Additionally, you can use an asterisk (*) as a wildcard if you want BUILDDB to add to the list of servers as it runs. This is a required parameter.

BUILDDB puts a collection trace into the BUILDDB.TRC file, and puts SQL statements that must be executed in DB2 into the BUILDDB.SQL file.

When you specify the wildcard character, *, in your replication network, BUILDDB examines all of the replication definitions for the servers provided and found in the *userid.*IBMSNAP_ROUTING, *userid.*IBMSNAP_REF_CNTL, *userid.*IBMSNAP_TRAIL, and *userid.*ASN. *timestamp*PC tables. The migration database contains global control tables that store registration and subscription details for these servers.

BUILDDB.SQL creates a new version of migration analysis data in your migration database every time you run it. The ASN.VERSION_CONTROL table contains the version and timestamp history of when you last executed BUILDDB.

Analyzing Data with PREPARE

Enter the following command to prepare the analysis of the migration database after you run the BUILDDB.SQL file:

```
ASNMIG PREPARE
mig_db ver x
```

Where:

mig_db Name of the migration database where the global control tables are stored. This is a required parameter.

Note: If you want to create your global migration control tables in a DB2 for MVS subsystem, DSNDB04 is the default database.

ver Version of the migration data that you want to analyze. You can use L to select the latest version, or you can select a specific version as noted in your migration database. This is a required parameter.

x Use either V or O in the position of this character. V validates only the replication definitions in the global control tables before proceeding with Version 5 migration. O updates the replication definitions in the global control tables based on the Version 5 control table data. This is an optional parameter.

PREPARE puts an analysis trace into the PREPARE.TRC file, and puts an analysis summary into the PREPARE.RPT file. The trace file is not translated, and should be handled like the Apply trace file. You should always save the output of each run, and carefully analyze the PREPARE.RPT file for any migration issues.

Migration issues that the PREPARE program checks for are:

- Subscription predicates that limit the scope of data applied
- Subscriptions released by EI SQL statements to the IBMSNAP.REF_CNTL table
- Subscriptions referencing Version 1 control tables
- Subscriptions using the FA run mode. All other run modes are converted to type AO
- Subscriptions scheduled with ENABLE = -n
- SQL_STMTs directed to run elsewhere from the source or target server
- Configurations requiring simultaneous Version 5 migration
- Grouping subscriptions in the replication subscription
- DataJoiner servers
- Existence of convergent consistent tables
- Join registrations through views

Migrating Data with MIGRATE

The MIGRATE program functions include:

- The MIGRATE program migrates one source server, one target server, or one of each (source and target) at a time.
- The output of the MIGRATE program puts a migration trace into the MIGRATE.TRC file, and a report of the success or failure of the migration execution in the MIGRATE.RPT file. If the migration fails, the changes will be restored to their original state according to the migration assumptions.

  The trace file is not translated, and should be handled like the Apply trace file. You should always save the output of each run, especially if you migrate the source server before the target servers. Otherwise, the source server trace is lost.
- The MIGRATE program reads the run-time related information, such as the timestamp, log sequence number, or status from source, target, and control servers during migration or fallback.
- When you migrate the Capture program, all registrations at the source server are migrated to Version 5. When you migrate the Apply program, all subscriptions at all the control servers in the subscriber ID's routing table are migrated to Version 5 subscriptions. The migrate program migrates a Version 1 multiple control server environment to one Version 5 control server. When you run ASNMIG, you must specify the location for the consolidated Version 5 control server.

The following sections explain how to migrate the Version 1 registrations and Version 1 subscriptions.

Migrating a Capture Program Process

To migrate the Capture program from Version 1 to Version 5:

1. Stop the Capture program.
2. Connect to the source server database by entering:

   ```
   DB2 CONNECT TO
   database
   ```

   where *database* is the source server database.
3. Bind the migration program to the migration database and all associated source, target, and control servers.
4. Enter the following command:

   ```
   ASNMIG MIGRATE D=
   mig_db S=
   v1_data_serv (V=
   ver)
   ```

   Where:

   mig_db Name of the migration database where the global control tables are stored. This is a required parameter.

v1_data_serv	Name of the Version 1 source server, known as the source server in Version 5. This is a required parameter.
ver	Version of the migration data that you want to migrate. If you do not specify this parameter, ASNMIG selects the latest version number. This is an optional parameter.

You can specify only one migration database, source server, and version.

5. Examine the migration report (MIGRATE.RPT) to determine if the program executed successfully.
6. Make sure the Version 5 Capture program is installed on the source server. To install the program, see *Quick Beginnings* for your particular platform.
7. Bind the Version 5 Capture program as described in "Configuring the Capture Program for Windows NT and Windows 95" on page 145 for Windows NT and Windows 95, and in "Configuring the Capture Program for OS/2" on page 167 for OS/2.
8. Warm start the Version 5 Capture program as described in "Starting Capture for Windows NT and Windows 95" on page 151 for Windows NT and Windows 95, and in "Starting Capture for OS/2" on page 171 for OS/2.

When you run ASNMIG MIGRATE for a source server where Capture might run, MIGRATE performs the following actions:

1. Connects to the source server that you specify.
2. Drops the critical section table, if an Apply program process has not locked it, and then creates it with the Version 5 Data Definition Language (DDL). If the critical section table is locked, ASNMIG attempts up to 400 times to drop it. There is one critical section table per registered source.
3. Creates the ASN.IBMSNAP_WARM_START table if it does not exist. Otherwise, the rows in all warm start tables are deleted. That is, all Version 1 and Version 5 warm start tables are empty after migration.
4. Alters the ASN.IBMSNAP_CCPPARMS table to add a new PRUNE_INTERVAL column.
5. Alters the ASN.IBMSNAP_UOW table to add two new columns: IBMSNAP_REJ_CODE and IBMSNAP_APPLY_QUAL.
6. Creates the Version 5 ASN.IBMSNAP_REGISTER and ASN.IBMSNAP_PRUNCNTL tables.
7. Creates the Version 1 ASN.IBMSNAP_CD_CNTL1 table for saving Version 1 registrations.
8. Copies from the Version 1 ASN.IBMSNAP_CD_CNTL and ASN.GLOBAL_CD_CNTL tables to the Version 5 ASN.IBMSNAP_REGISTER table.
9. Copies from the Version 1 ASN.IBMSNAP_CD_CNTL table to the ASN.IBMSNAP_CD_CNTL1 table.

10. For each registered source:
    a. If the length of the pruning control table name is 18 characters, MIGRATE changes the first character of the pruning control table name to “M.” If the length of the pruning control table name is less than 18 characters, creates a backup Version 1 pruning control table with a 1 appended to the original pruning control table name.
    b. Copies from the original pruning control table to the backup table individually.
    c. Copies the Version 1 pruning control table to the Version 5 pruning control table.
    d. Drops the Version 1 pruning control tables.
    e. Creates the Version 1 pruning control view based on the Version 5 pruning control table.
    f. Copies the SOURCE_OWNER, SOURCE_TABLE, and SOURCE_VIEW_QUAL values for a view registered source from the ASN.GLOBAL_CD_CNTL table. They are resolved in the BUILDDB and PREPARE actions.
11. Drops the Version 1 ASN.IBMSNAP_CD_CNTL table, and then creates the Version 1 ASN.IBMSNAP_CD_CNTL view based on the join of entries in the Version 1 ASN.IBMSNAP_CD_CNTL1 and Version 5 ASM.IBMSNAP_REGISTER tables. It gets the names of the pruning control view from ASN.IBMSNAP_CD_CNTL1 and the critical section name as a constant.

Migrating an Apply Program Process

To migrate Version 1 subscriptions to Version 5:

1. Stop the Version 1 Apply instances currently executing on the target servers to be migrated.
2. Connect to the target server database by entering:

   ```
   DB2 CONNECT TO
   database
   ```

   where *database* is the target server database.
3. Bind the migration program, if you have not done so, to the migration database and all associated source, target, and control servers.
4. Enter the following command:

   ```
   ASNMIG MIGRATE D=
   mig_db T=
   rout_serv C=
   v5_ctrl_serv U=
   sub_id (V=
   ver)
   ```

   Where:

mig_db	Name of the migration database where the global control tables are stored. This is a required parameter.
rout_serv	Server where the Version 1 routing table resides. This is a required parameter.
v5_ctrl_serv	Version 5 control server, which contains the control tables for the Apply program process. This is a required parameter.
sub_id	Subscriber ID of an Apply program process. This is a required parameter.
ver	Version of the migration data that you want to migrate. If you do not specify this parameter, ASNMIG selects the latest version number. This is an optional parameter.

You can specify only one control server, subscriber ID, migration database where the routing table server resides, and version.

5. Examine the migration report (MIGRATE.RPT) to determine if the program executed successfully.
6. Make sure the Version 5 Apply program is installed on the correct server. To install the program, see *Quick Beginnings* for your particular platform.
7. Bind the Version 5 Apply program as described in “Configuring the Apply Program for Windows NT and Windows 95” on page 146 for Windows NT and Windows 95, and in “Configuring the Apply Program for OS/2” on page 168 for OS/2.
8. Start the Version 5 Apply program as described in “Starting Apply for Windows NT and Windows 95” on page 157 for Windows NT and Windows 95, and in “Before You Start theApply Program” on page 177 for OS/2.

When you run ASNMIG MIGRATE for a target server where Apply might run, MIGRATE performs the following actions:

1. Connects to the Version 5 control server.
2. Creates the following Version 5 common subscription control tables at the Version 5 control server:
   - ASN.IBMSNAP_SUBS_SET
   - ASN.IBMSNAP_SUBS_MEMBR
   - ASN.IBMSNAP_SUBS_COLS
   - ASN.IBMSNAP_SUBS_STMTS
   - ASN.IBMSNAP_SUBS_EVENT
   - ASN.IBMSNAP_APPLYTRAIL
3. Connects to the server where the routing table resides, and retrieves all Version 1 control server names.
4. For each Version 1 control server:
   a. Copies the Version 1 *userid*.IBMSNAP_REF_CNTL table at the Version 1 control server to the Version 5 ASN.IBMSNAP_SUBS_SET, ASN.IBMSNAP_SUBS_MEMBR, and ASN.IBMSNAP_SUBS_STMTS tables at the IBM Replication control server.

b. Copies the Version 1 *userid*.IBMSNAP_REF_COLS table at the Version 1 control server to the Version 5 ASN.IBMSNAP_SUBS_COLS table at the Version 5 control server.

c. Copies the Version 1 *userid*.IBMSNAP_TRAIL tables at the Version 1 control server to the Version 5 ASN.IBMSNAP_APPLYTRAIL table at the Version 5 control server.

5. If more than one Version 1 control server exists, the MIGRATE action collapses multiple Version 1 control servers into a single Version 5 control server. The subscription definitions from each Version 1 control server are copied to the single IBM Replication control server.
6. If auto-registration is detected at the target server, MIGRATE instructs you to enter the following command to migrate the ASN.IBMSNAP_CD_CNTL and *userid*.ASN. *timestamp*PC tables before the migration of this Apply program process is complete:

   ```
   ASNMIG MIGRATE D=
   mig_db S=
   v1_copy_serv (V=
   ver)
   ```

   Auto-registration occurs when the subscriber's COPY_OWNER and COPY_TABLE in the *userid*.IBMSNAP_REF_CNTL table are the same as the SOURCE_OWNER and SOURCE_TABLE in the ASN.IBMSNAP_REGISTER table at the target server.
7. Creates *userid*.IBMSNAP_ROUTING1, copies data from userid.IBMSNAP_ROUTING to it, then drops *userid*.IBMSNAP_ROUTING. This prevents you from inadvertently running the Version 1 Apply program process after you migrate to Version 5.

Reverting to Version 1 with FALLBACK

You can revert to Version 1 by running the ASNMIG command with the FALLBACK action. You need to fall back from all Version 5 Apply program processes before you fall back from associated Version 5 Capture program processes. You can fall back from Apply program processes in whichever order you choose.

After you fall back to Version 1, you can use the same migration process as described in “Migrating Data with MIGRATE” on page 256. You cannot fall back to Version 1 after executing the CLEANUP action, because all unused Version 1 control tables and views are dropped.

When fallback occurs at the source server, ASN.IBMSNAP_CRITSEC, ASN.IBMSNAP_UOW, and ASN.IBMSNAP_CCPPARMS remain in their altered format with the additional columns, which do not cause any conflicts with the Version 1 Capture and Apply programs.

If you successfully migrate both Capture and Apply and you subsequently used the Control Center to define new replication sources and subscriptions before falling back, all new Version 5 replication sources and subscriptions are lost. When this happens, the migration program issues an informational message.

Falling Back from an Apply Program Process

To fall back from Version 5 to Version 1 for an Apply program process:

1. Stop the Version 5 Apply program that you are migrating.
2. Enter the following command:

   ```
   ASNMIG FALLBACK D=
   mig_db T=
   rout_serv C=
   v5_ctrl_serv U=
   sub_id (V=
   ver)
   ```

   Where:

   mig_db Name of the migration database where the global control tables are stored. This is a required parameter.

   rout_serv Server where the routing table resides. This is a required parameter.

   v5_ctrl_serv Version 5 control server, which contains the control tables for the Apply program process. This is a required parameter.

   sub_id Subscriber ID of an Apply program process. This is a required parameter.

   ver Version of the migration data that you want to fall back to. If you do not specify this parameter, ASNMIG selects the latest version number. This is an optional parameter.

3. Examine the fallback report (FALLBACK.RPT) to determine if the program executed successfully.
4. Make sure the Version 1 Apply program still exists on the source server where it will execute. To install the program, see *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform.
5. Bind the Version 1 Apply program as described in *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform.
6. Start the Version 1 Apply program as described in *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform after you successfully execute the FALLBACK action.

When you start ASNMIG FALLBACK for a target server, the program performs the following actions:

1. Creates *userid*.IBMSNAP_ROUTING, copies data from *userid*.IBMSNAP_ROUTING1 to it, then drops *userid*.IBMSNAP_ROUTING1.
2. Copies back the information in the Version 5 common subscription tables to the IBMSNAP_REF_CNTL and IBMSNAP_TRAIL tables at the Version 1 control

server. It matches the Version 1 DATA_SERVER, BASE_OWNER and table, COPY_SERVER, and COPY_OWNER and table to the Version 5 SOURCE_SERVER, SOURCE_OWNER and table, TARGET_SERVER, and TARGET_OWNER and table, respectively.

If auto-registration is detected at the target server, FALLBACK instructs you to enter the following command to fall back the ASN.IBMSNAP_REGISTER and *userid.*ASN. *timestamp*PC tables after the fallback of this Apply program process is complete:

```
ASNMIG FALLBACK D=
mig_db S=
v5_targ_serv (V=
ver)
```

Auto-registration occurs when the TARGET_OWNER and TARGET_TABLE columns in the subscriber's ASN.IBMSNAP.SUBS_MEMBR table are the same as the SOURCE_OWNER and SOURCE_TABLE columns in the ASN.IBMSNAP_REGISTER table at the target server.

You can fall back the ASN.IBMSNAP_REGISTER and *userid.*ASN. *timestamp*PC tables after the fallback of this Apply program process is finished.

Falling Back from a Capture Program Process

To fall back from Version 5 to Version 1 for a Capture program process:

1. Stop the Capture program.
2. Enter the following command:

   ```
   ASNMIG FALLBACK D=
   mig_db S=
   source_serv (V=
   ver)
   ```

   Where:

   mig_db Name of the migration database where the global control tables are stored. This is a required parameter.

   source_serv Source server of Version 5. This is a required parameter.

   ver Version of the migration data that you want to fall back to. If you do not specify this parameter, ASNMIG selects the latest version number. This is an optional parameter.

3. Examine the fallback report (FALLBACK.RPT) to determine if the program executed successfully.
4. Make sure the Version 1 Capture program is installed on the source server. To install the program, see *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform.
5. Bind the Version 1 Capture program as described in *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform.

6. Warm start the Version 1 Capture program as described in *DataPropagator Relational Guide (SC26-3399-05)* for your particular platform after you successfully execute the FALLBACK action.

When you start ASNMIG FALLBACK for a source server, the program performs the following actions:

1. Drops the ASN.IBMSNAP_CD_CNTL view.
2. Creates the ASN.IBMSNAP_CD_CNTL table, then copies data to it from the Version 1 ASN.IBMSNAP_CD_CNTL1 and Version 5 ASN.IBMSNAP_REGISTER tables.
3. Drops the ASN.IBMSNAP_CD_CNTL1 table.
4. Drops the Version 1 *userid.*ASN. *timestamp*PC views and creates the Version 1 *userid.*ASN. *timestamp*PC tables.
5. Copies data from the Version 1 backup pruning control tables and the Version 5 *userid.*ASN. *timestamp*PC table to the Version 1 pruning control tables.

Dropping Version 1 Control Tables and Migration Control Tables with CLEANUP

If all replication components in a network are migrated and perform satisfactorily, you can run the ASNMIG command with the CLEANUP action to drop Version 1 views and tables at the Version 1 control, source, and target servers. Before you enter the command, be aware that you cannot fall back after entering the command.

Enter the following command to clean up the Version 1 views and tables:

```
ASNMIG CLEANUP D=
mig_db S=
data_serv C=
v5_ctrl_serv T=
rout_serv U=
sub_id (V=
ver)
```

Where:

mig_db Name of the migration database where the global control tables are stored. This is a required parameter.

data_serv Version 1 source server. This is a required parameter.

v5_ctrl_serv Version 5 control server. This is a required parameter.

rout_serv Server where the routing table resides. This is a required parameter.

sub_id Subscriber ID of an Apply program process. This is a required parameter.

ver Version of the migration data that you want to clean up. If you do not specify this parameter, ASNMIG CLEANUP selects the latest version number. This is an optional parameter.

After you enter this command, the following tables or views are dropped:

- At the Version 1 source server:
  - ASN.IBMSNAP_CD_CNTL1 table
  - ASN.IBMSNAP_CD_CNTL view
  - Migration pruning control views and backup pruning control tables
  - ASN.IBMSNAP_AUTH table
- At the Version 1 control server:
  - *subscriber*.IBMSNAP_REF_CNTL tables
  - *userid*.IBMSNAP_REF_COLS tables
  - *userid*.IBMSNAP_TRAIL tables
- At the Version 1 target server (Version 5 target server):
  - *userid*.IBMSNAP_ROUTING1 tables
  - ASN.IBMSNAP_AUTH table
  - ASN.IBMSNAP_CD_CNTL1 table, ASN.IBMSNAP_CD_CNTL view, and pruning control views if auto-registration occurs

Chapter 20. Table Structures

This chapter describes the relational database tables that IBM Replication uses in its copy operations.

Sample CREATE TABLE statements are provided for all tables.

Table 20, Table 21, and Table 22 on page 266 provide directories to the tables described in this chapter. These tables list the table names, the internal names, and the reference page number.

Table 20. Quick Reference for Tables Used at the Source Server during IBM Replication Processes

Table name	Internal name	Refer to
Capture enqueue table	ASN.IBMSNAP_CCPENQ	268
Change data table	userid.ASNtimestampCD	268
Critical section table	ASN.IBMSNAP_CRITSEC	270
Pruning control table	ASN.IBMSNAP_PRUNCNTL	270
Register table	ASN.IBMSNAP_REGISTER	273
Trace table	ASN.IBMSNAP_TRACE	277
Tuning parameters table	ASN.IBMSNAP_CCPPARMS	278
Unit-of-work table	ASN.IBMSNAP_UOW	279
Warm start table	ASN.IBMSNAP_WARM_START	281
Warm start table for Capture for VSE and VM	ASN.IBMSNAP_WARM_START	282

Table 21. Quick Reference for Tables Created at the Control Server during IBM Replication Processes

Table name	Internal name	Refer to
Apply trail table	ASN.IBMSNAP_APPLYTRAIL	284
Subscription columns table	ASN.IBMSNAP_SUBS_COLS	287
Subscription events table	ASN.IBMSNAP_SUBS_EVENT	288
Subscription set table	ASN.IBMSNAP_SUBS_SET	289
Subscription statements table	ASN.IBMSNAP_SUBS_STMTS	292
Subscription targets member table	ASN.IBMSNAP_SUBS_MEMBR	294

Table 22. Quick Reference for Target Tables

Table name	Internal name	Refer to
Base aggregate target table	userid.target_table	297
Change aggregate target table	userid.target_table	297
Consistent change data target table	userid.target_table	298
Point-in-time target table	userid.target_table	300
Replica target table	userid.target_table	301
User copy target table	userid.target_table	302

Control Tables Used at the Source Server

Table 23 on page 267 lists the tables that are used at the control server and their internal names and provides a brief description of each table. The structure of all control tables is subject to change.

Table 23. Control Tables Used at the Source Server

Table name	Internal name	Description
Capture enqueue table	ASN.IBMSNAP_CCPENQ	
Change data table	*userid*.ASNtimestampCD	This table is created while defining replication sources. It contains changed data information. The timestamp in the table name is from the source server; for auto-registered tables, the timestamp is from the target server.
Consistent change data table	userid.target_table	There are two types of consistent change data tables; internal and external. The internal table is a join of the change data and UOW tables. An external consistent change data table is also called an external source table because it originates outside the IBM Replication environment. Any table created outside that environment that conforms to the consistent change data table format is an external source table.
Critical section table	ASN.IBMSNAP_CRITSEC	This table is used as a logical lock between the Capture program and the Apply program for serialization of internal logic.
Pruning control table	ASN.IBMSNAP_PRUNCNTL	This table contains pruning control information for the current source server. There is one at each source server and one row per source-to-target copy. It translates time stamp/log address, allows the Apply program to trigger the Capture program, and coordinates pruning.
Register table	ASN.IBMSNAP_REGISTER	This table contains information about replication sources, such as the names of the replication source tables, their attributes, and their staging table names.
Trace table	ASN.IBMSNAP_TRACE	This table is required if the Capture program is installed. It contains Capture program audit trail information.
Tuning parameters table	ASN.IBMSNAP_CCPPARMS	This table contains parameters that can be modified to control the performance of the Capture program.
Unit-of-work table	ASN.IBMSNAP_UOW	This table is required for transaction consistency. It is a staging table that contains commit information. The information from this table is joined with change information from the change data table to produce consistent, committed changes.
Warm start table	ASN.IBMSNAP_WARM_START	This table contains information that enables the Capture program to resynchronize.
Warm start table for Capture for VSE and VM	ASN.IBMSNAP_WARM_START	This table contains information that enables the Capture program to resynchronize in the VM and VSE environments.

Diagnosis, Modification or Tuning information

Capture Enqueue Table

- ASN.IBMSNAP_CCPENQ

The Capture enqueue table is used in the MV and VSE environments only.

Table 25 on page 269 provides a list and a brief description of the Capture enqueue table column.

Table 24. Capture Enque Table Column

Column name	Description
LOCKNAME	The unique name of the resource for this database.

Figure 34 on page 269 shows the index for Capture enqueue tables.

```
CREATE TABLE useridIBMSNAP_CCPENQ
 LOCKNAME CHAR(9) NOT NULL);
```

Figure 33. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Product-sensitive programming interface

Change Data Table

- userid.ASNtimestampCD

CD tables contain changed data assigned to source table rows, in time series. The changed data is used to update other target tables. There is one CD table for each base table, created while defining a replication source that is enabled for data capture.

CD tables can contain both committed and uncommitted changes, and possibly incomplete changes in rows.

Commit sequencing, which is not known when records are inserted into the CD table, is provided through the unit-of-work table.

Normally, the Capture program inserts rows into this table for a particular platform. It also deletes rows periodically.

The Control Center automatically creates the indexes. A non-unique descending index is required on IBMSNAP_UOW.

Other qualities of CD tables are as follows:

- Before-image user data columns and computed user data columns should be nullable.
- Null attributes of the after-image user data columns should match null attributes of the source.
- Views of change data tables can be included in view replication sources.
- If standard or enhanced conflict detection is used for replicas, all before-image and after-image column values must be captured.
- DPROPR V1 supported an IBMSNAP_LOGMARKER column in the changed data tables. For compatibility with existing installations, the presence of the once optional IBMSNAP_LOGMARKER column is tolerated. This column is no longer used.

Table 25 provides a list and a brief description of each of the CD table columns.

Table 25. CD Table Columns

Column name	Description
IBMSNAP_UOW	Unit-of-work ID. It is also the foreign key into the ASN.IBMSNAP_COMMIT table.
IBMSNAP_INTENTSEQ	Unique identifier for this change; it describes the sequence of a change within a transaction.
IBMSNAP_OPERATION	Character value of I, U, or D, indicating an insert, update, or delete record.
<user data columns>	Columns from source tables specified while defining replication sources.

Figure 34 shows the index for CD tables.

```
CREATE TABLE CD (
 IBMSNAP_UOWID CHAR(10) FOR BIT DATA  NOT NULL,
 IBMSNAP_INTENTSEQ CHAR(10) FOR BIT DATA  NOT NULL,
 IBMSNAP_OPERATION CHAR(1) NOT NULL,
 KEY1 CHAR(1),
 DATA1 CHAR(1) )
CREATE INDEX ZCD ON CD(IBMSNAP_UOWID DESC);
```

Figure 34. CREATE TABLE Statement

End of Product-sensitive programming interface

Diagnosis, Modification or Tuning information

Critical Section Table

- ASN.IBMSNAP_CRITSEC

The critical section table is used for concurrence control purposes between the Capture and Apply programs.

Table 26 provides a list and a brief description of each of the CCD table columns.

Table 26. Critical Section Table Columns

Column name	Description
APPLY_QUAL	The unique value used to associate each replication subscription to an Apply instance in the **Apply qualifier** field. This value must be supplied by the user during subscription. This value is used to populate the IBMSNAP_APPLY_QUAL column of the IBMSNAP_UOW table.

Figure 35 shows the index for critical section tables.

```
CREATE TABLE ASN.IBMSNAP_CRITSEC(
 APPLY_QUAL CHAR(18) NOT NULL,
 <+padding where row level locking not supported>)
 DATA CAPTURE CHANGES;
CREATE UNIQUE INDEX IBMSNAP_CRITSECX ON ASN.IBMSNAP_CRITSEC (APPLY_QUAL);
```

Figure 35. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Product-sensitive programming interface

Pruning Control Table

- ASN.IBMSNAP_PRUNCNTL

The pruning control table provides a timestamp and log-address translation in case the SYNCHPOINT column for Capture for MVS is a relative byte address value.

Also, the pruning control table coordinates the pruning of the change data tables, which have the potential for unlimited growth. There is one pruning control table at each source server and one row for each subscription member.

This table is located at the source server, but it is also automatically created at the target server when replica and CCD target tables are defined. Replica and CCD target tables created at the target server can be used as source tables for further subscriptions, making the target server a source server as well. This is performed as part of auto-registration, described in Part 3, “Administering Your Replication System” on page 81

The rows in the pruning control table are not deleted during a cold start of the Capture program. The Control Center uses the values from the pruning control table to list direct copies of a particular source table.

Table 27 provides a list and a brief description of each of the pruning control table columns.

Table 27 (Page 1 of 2). Pruning Control Table Columns

Column name	Description
TARGET_SERVER	The target server to the Apply program updating this row; also, the name of the database (DRDA application server) where the target is stored. For DB2 for OS/2, there is no DRDA application server, so this column is informational. Matches the column in the IBMSNAP_SUBS_SET table of the same name.
TARGET_OWNER	The middle qualifier of the target table name; default is the user ID of the user defining the subscription. Matches the column in the IBMSNAP_SUBS_MEMBR table of the same name.
TARGET_TABLE	The name of the target table as defined in the subscription default as the source server name. Matches the column in the IBMSNAP_SUBS_MEMBR table of the same name.
SYNCHTIME	A source server timestamp, which can be added to any captured log records, for any source table, indicating that the change did not occur before this time. If the log records are individually timestamped, use those timestamps; otherwise, these values are approximate and are set by the Apply program at the start of a subscription cycle and after each subsequent cycle. They are eventually appended into each change data table row.
SYNCHPOINT	The SYNCHPOINT value equals the SYNCHPOINT field value in the common subscription set table. This value is used to coordinate the pruning of change data tables. The Apply program sets this initial value to 0, indicating refresh. If the Apply program sets a nonzero value, the change data table can be eligible for pruning.
SOURCE_OWNER	The middle-level qualifier of the source table.
SOURCE_TABLE	The name of the source table.
SOURCE_VIEW_QUAL	Supports join subscriptions by matching the similar columns in ASN.IBMSNAP_REGISTER. Without this column, it would not be possible to support multiple subscriptions for the different source views with identical SOURCE_OWNER, SOURCE_TABLE values.
APPLY_QUAL	Part of the foreign key from the IBMSNAP_SUBS_SET table.
SET_NAME	Part of the foreign key from the IBMSNAP_SUBS_SET table.
CNTL_SERVER	The RDB name of the control server for the Apply programupdating this row; either Apply for MVS, VSE, or VM.

Table 27 (Page 2 of 2). Pruning Control Table Columns

Column name	Description
TARGET_STRUCTURE	A value that identifies the type of target table: **1** Source table **2** N/A **3** CCD table **4** Point-in-time table **5** Base aggregate table **6** Change aggregate table **7** Replica **8** User copy
CNTL_ALIAS	The name of the control server used by the Apply program on the DB2 Universal Database. This identifies the location of the subscription definition and is later used to perform administrative actions on a subscription. This does not necessarily match the alias of the control server used by the Apply program for the DB2 Universal Database.

Figure 36 shows the index for pruning control tables.

```
CREATE TABLE ASN.IBMSNAP_PRUNCNTL (
 TARGET_SERVER CHAR (18) NOT NULL,
 TARGET_OWNER CHAR (18) NOT NULL,
 TARGET_TABLE CHAR(18) NOT NULL,
 SYNCHTIME TIMESTAMP
 SYNCHPOINT CHAR(10) FOR BIT DATA,
 SOURCE_OWNER CHAR (18),
 SOURCE_TABLE CHAR (18),
 SOURCE_VIEW_QUAL SMALLINT,
 APPLY_QUAL CHAR (18),
 SET_NAME CHAR (18),
 CNTL_SERVER CHAR(18) NOT NULL,
 TARGET_STRUCTURE SMALLINT NOT NULL,
 CNTL_ALIAS CHAR(8)
 < + padding where row level locking not supported>)
  DATA CAPTURE CHANGES;
CREATE UNIQUE INDEX IBMSNAP_PRUNCNTLX ON
ASN.IBMSNAP_PRUNCNTL (SOURCE_OWNER, SOURCE_TABLE,
SOURCE_VIEW_QUAL, SET NAME,TARGET_SERVER, TARGET_TABLE,
TARGET_OWNER);
```

Figure 36. CREATE TABLE Statement

To protect against the potential for clone descriptions which differ only by the APPLY_QUAL value, that value is not included in the index and each entry is required to be posted at the time of subscription definition.

End of Product-sensitive programming interface

Product-sensitive programming interface

Register Table

- ASN.IBMSNAP_REGISTER

The register table holds information about Replication Sources objects at the source server.

A unique index is automatically created on columns SOURCE_OWNER and SOURCE_TABLE.

Table 28 (Page 1 of 4). Register Table Columns

SOURCE_OWNER	The owner of the source table whose updates are being captured.
SOURCE_TABLE	The name of the source table whose updates are being captured.
SOURCE_VIEW_QUAL	This value is set to equal 0 for registered physical tables and is greater than 0 for registered views.
GLOBAL RECORD	A flag that indicates whether this row is the global record. This record is maintained by the Capture program. If the Capture program is not installed, then there is no global record. **Y** This row is the global record. **N** This row is not the global record.
SOURCE_STRUCTURE	The structure of the base table: **1** source table **3** CCD table **4** point-in-time copy **5** base aggregate copy **6** change aggregate copy **7** replica **8** user copy
SOURCE_CONDENSED	A flag indicating: **Y** Changes can be netted out, with at most one row in the base table for every original table primary key value. **N** All changes must remain, retaining a complete update history. **A** Valid only for base aggregate or change aggregate tables.
SOURCE_COMPLETE	A flag indicating: **Y** The base table contains a row for every primary key value of interest. **N** The base table contains some subset of rows of primary key values.
CD_OWNER	The owner of the change data table.
CHANGE_DATA_TABLE	The name of the table for captured updates to the base table (set while defining the replication source).

Table 28 (Page 2 of 4). Register Table Columns

PHYS_CHANGE_OWNER	The owner of the PHYS_CHANGE_TABLE. For a view replication source, the value equals the value of the replication source that includes the change data referenced in the change data view definition. Capture uses this value to properly maintain CD_OLD_SYNCHPOINT and CD_NEW_SYNCHPOINT for view replication sources. The Apply program uses this value to properly maintain CCD_OLD_SYNCHPOINT and SYNCHPOINT for view repgistrations based on CCD copy table registrations that the Apply program maintains.
PHYS_CHANGE_TABLE	The name of the physical CD or CCD table. For a view replication source, the value equals the value of the replication source definition that includes the change data table referenced in the change data view definition. Capture uses this value to properly maintain CD_OLD_SYNCHPOINT and CD_NEW_SYNCHPOINT for view replication sources. The Apply program uses this value to properly maintain CCD_OLD_SYNCHPOINT and SYNCHPOINT for view registrations based on the CCD copy table the Apply program maintains.
CD_OLD_SYNCHPOINT	The SYNCHPOINT value of the oldest row in the change data table. The Capture program sets this value when the change data table is emptied. If null, the Capture program sets an initial value (the RBA of the logmarker timestamp related to a refresh of the associated base table). You define whether the Capture and Apply programs set subsequent values when old rows are pruned from the CD table.
CD_NEW_SYNCHPOINT	The SYNCHPOINT value associated with the most recent change inserted into the CHANGE DATA TABLE. If the Capture program has not inserted into the CHANGE DATA TABLE recently, then the value will not advance. **Note:** This value reduces the need to prepare and open cursors on change data tables. CD_NEW_SYNCHPOINT can only reliably indicate a *no-change* condition, when CD_NEW_SYNCHPOINT is less than or equal to IBMSNAP_SUBS_SET.SYNCHPOINT. Change conditions are not reported as reliably because uncommitted work will cause this value to increase, even when there are no committed updates to replicate.
DISABLE_REFRESH	A flag that indicates whether refresh queries are issued against the source table if a change data table becomes invalid, if a gap is detected, or if the Capture program cold starts. This prevents refresh activity from overloading the source database when the Capture program is restarted. You can set this flag manually, use a program at the source database site to set it, or have the Capture program set it automatically while executing an UPDATE operation. **0** Full refreshes are enabled. **1** Full refreshes are prevented. This column is initialized to 1.
CCD_OWNER	The owner of the local consistent change data table.
CONSISTENT CHANGE DATA TABLE	The name of the staging table that contains committed-only captured updates as copied from a join of the local change data table (defined by a subscription definition, which names the source table and the unit-of-work table).

Table 28 (Page 3 of 4). Register Table Columns

CCD_OLD_SYNCHPOINT	The SYNCHPOINT value of the oldest row in the external consistent change data table. This value is set when one of the applications listed below is used to generate the CONSISTENT CHANGE DATA TABLE. CCD_OLD_SYNCHPOINT is set in the following ways: • By the Control Center when the consistent change data table is being auto-registered. CCD_OLD_SYNCHPOINT is set to NULL. • By the Control Center when a consistent change data table is defined as an external replication source table. CCD_OLD_SYNCHPOINT is set to MIN(IBMSNAP_COMMITSEQ) of the CCD_TABLE. • Otherwise by the Apply program or another external application.
SYNCHPOINT	The sequence value that is useful for maintaining the state of consistent change data table copies, subscription states, and for controlling pruning.
SYNCHTIME	The timestamp associated with the SYNCHPOINT column.
CCD_CONDENSED	A flag indicating: **Y** Changes can be netted out, with at most one row in the CCD table for every base table primary key value. **N** All changes must remain, retaining a complete update history.
CCD_COMPLETE	A flag indicating that: **Y** The CCD table contains a row for every primary key value of interest. **N** The CCD table contains some subset of rows of primary key values.
ARCH_LEVEL	The architectural level of the definition in the row. This level is defined by IBM, and for Version 5 is '0201'.
DESCRIPTION	A field for user comments entered while defining replication sources.
BEFORE_IMG_PREFIX	Represents the default character identifying before-image column names in the CD table. The value can be null, but must not match any leading character identifying after-image user data column names in the CD table. The length of PREFIX_CHAR is as follows: **1** For an ASCII single-byte character system prefix character. **1** For an EBCDIC single-byte character system prefix character. **2** For an ASCII double-byte character system prefix character. **4** For an EBCDIC double-byte character system prefix character. This allows for shift-in and shift-out characters.
CONFLICT_LEVEL	Where: **2** Enhanced detection with cascading transaction rejection. **1** Standard detection (no log flush to CD) with cascading transaction rejection. **0** No conflict detection. **Note:** • CONFLICT_LEVEL 2 is only supported for DB2 for MVS version 4 or higher. • CONFLICT_LEVEL is assumed to never change and to be the same for all descendents of the user table.

Table 28 (Page 4 of 4). Register Table Columns

PARTITION _KEYS_CHG	This value is assumed to be the same for all the user table's dependent replics. Where:	
	N	The source table columns are:Not updated, if part of the primary key of a copy table Not updated, if part of a target table space partitioning key (DB2 for MVS) or node positioning key (DB2 Parallel Edition) Not updated, if referenced in any subscription predicate
	Y	The columns in the SOURCE_TABLE that are included in the partitioning keys are never ever *updated* by applications. *Updates* are captured as two rows in the change data table with ascending IBMSNAP_INTENTSEQ values. The first row contains the IBMSNAP_OPERATION value, where 'D' = all other deletions, meaning the before-image of all columns. The second row contains the IBMSNAP_OPERATION value, where 'I' = all other insertions, meaning the after-image of all columns. The delete must have a lower IBMSNAP_INTENTSEQ value that the insert, so that the updated partitioning key will not allow out-migration from a particular partition. In such cases, the insert must come last so that the row will remain in the partition.
	NULL	If this is the global control row.

```
CREATE TABLE ASN.IBMSNAP_REGISTER (
  SOURCE_OWNER CHAR(18) NOT NULL,
 SOURCE_TABLE CHAR(18) NOT NULL,
 SOURCE_VIEW_QUAL SMALLINT NOT NULL,
 GLOBAL_RECORD CHAR(1) NOT NULL,
 SOURCE_STRUCTURE SMALLINT NOT NULL,
 SOURCE_CONDENSED CHAR(1) NOT NULL,
 SOURCE_COMPLETE CHAR(1) NOT NULL,
 CD_OWNER CHAR(18),
 CD_TABLE CHAR(18),
 PHYS_CHANGE_OWNER CHAR(18),
 PHYS_CHANGE_TABLE CHAR(18),
 CD_OLD_SYNCHPOINT CHAR(10) FOR BIT DATA,
 CD_NEW_SYNCHPOINT CHAR(10) FOR BIT DATA,
 DISABLE_REFRESH SMALLINT NOT NULL,
 CCD_OWNER CHAR(18),
 CD_TABLE CHAR(18),
 CCD_OLD_SYNCHPOINT CHAR(10) FOR BIT DATA,
 SYNCHPOINT CHAR(10) FOR BIT DATA,
 SYNCHTIME TIMESTAMP,
 CCD_CONDENSED CHAR(1),
 CCD_COMPLETE CHAR(1),
 ARCH_LEVEL CHAR(4) NOT NULL,
 DESCRIPTION CHAR(254),
 BEFORE_IMG_PREFIX VARCHAR(4),
 CONFLICT_LEVEL CHAR(1),
 PARTITION_KEYS_CHG CHAR(1),
 <+padding where row level locking not supported>);
CREATE UNIQUE INDEX ZIBMSNAP_REGISTERX ON
 ASN.IBMSNAP_REGISTER (SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL);
```

Figure 37. CREATE TABLE Statement

End of Product-sensitive programming interface

Diagnosis, Modification or Tuning information

Trace Table

- ASN.IBMSNAP_TRACE

This table contains trace information for Capture program.

Note:

- On cold start, all of this table's entries will be deleted. On warm start, it will keep adding rows in a round robin fashion until the limit has been reached.

- The message numbers ASN0100I, ASN0101W, ASN0102W are issued for warnings and initialization information.

Table 29. Trace Table Columns

Column name	Description
OPERATION	The type of Capture operation, for example, initialization, capture, or error condition.
TRACE_TIME	Notes the time a row is inserted into the trace table.
DESCRIPTION	The message ID followed by the message text. The message can be informational or error. This column contains English-only text and is used by IBM service personnel.

```
CREATE TABLE ASN.IBMSNAP_TRACE (
 OPERATION CHAR(8) NOT NULL,
 TRACE_TIME TIMESTAMP NOT NULL,
 DESCRIPTION VARCHAR(254) NOT NULL);
CREATE INDEX ASN.ZIBMSNAP_TRACE
 ON ASN.IBMSNAP_TRACE (TRACE_TIME ASC)
```

Figure 38. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Diagnosis, Modification or Tuning information

Tuning Parameters Table

- ASN.IBMSNAP_CCPPARMS

This table contains parameters that you can modify to control the performance of the Capture program.

Table 30 (Page 1 of 2). Tuning Parameters Table Columns

Column name	Description
RETENTION_LIMIT	The age limit, in minutes, for keeping CD table rows. This value is used with PRUNE_LIMIT_SEQ to determine the pruning limit. Any change data rows older than this value are pruned, even if they have not been copied by all clients. The default value is 10,800.
LAG_LIMIT	The amount of time, in minutes, that Capture program is allowed to lag in processing log records before it shuts itself down. During periods of high update frequency, refreshes can be more economical than updates. The default value is 10,800.

Table 30 (Page 2 of 2). Tuning Parameters Table Columns

Column name	Description
COMMIT_INTERVAL	The Capture program commit threshold, in seconds, for any inserts, updates, or deletes to any CD tables, pruning control tables, and the global ASN.IBMSNAP_UOW table. The default value is 30. On systems that do not support ISOLATION (UR), this value should be less than the DB2 lock timeout value, to prevent Apply program instances from timing out due to contention with the Capture program.
PRUNE_INTERVAL	The Capture program commit threshold, in seconds, for automatic deletion of CD and UOW rows that are no longer needed. The default value is 300. Values set lower will save space, but will increase processing costs. Values set higher require more CD and UOW table space but decrease processing costs.

```
CREATE TABLE ASN.IBMSNAP_CCPPARMS (
 RETENTION_LIMIT INT,
 LAG_LIMIT INT,
 COMMIT_INTERVAL INT,
 PRUNE_INTERVAL INT)
```

Figure 39. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Product-sensitive programming interface

Unit-of-Work Table

- ASN.IBMSNAP_UOW

The unit-of-work (UOW) table ensures data integrity by recording transactions that have been committed at the source server. By performing a join of this table and the changes logged by the Capture program, the Apply program ensures that only committed changes are being copied.

This table is indexed into the change data (CD) table, ordered by the commit sequence. This order is necessary for the following reasons:

- The CD table is assumed to contain uncommitted changes.
- The CD table cannot be ordered in commit sequence without updating records after insertion, because the relative timing of row updates might not be reflected in the relative timing of commits.

Note: This table is created automatically during the define replication sources process. The existence and use of this table are required if the Capture program is installed.

Table 31. UOW Table Columns

Column name	Description
IBMSNAP_UOWID	The unit-of-recovery ID from the log record header for this unit-of-work.
IBMSNAP_COMMITSEQ	The transaction commit sequencing value.
IBMSNAP_LOGMARKER	The approximate commit time at the source server.
IBMSNAP_AUTHTKN	The authorization token associated with the transaction. It is useful for database auditing. For DB2 for MVS, this field is the correlation ID. This column is not automatically copied to other tables; you must select it and copy it as a user data column.
IBMSNAP_AUTHID	The authorization ID associated with the transaction. It is useful for database auditing. For DB2 for MVS, this field is the primary authorization ID. This column is not automatically copied to other tables; you must select it and copy it as a user data column.
IBMSNAP_REJ_CODE	The character value. **0** for a transaction with no known conflict **1** for a transaction containing at least one same-row intra-table update conflict **2** for a cascade-rejection of a transaction dependent on a prior transaction having at least one same-row conflict **3** for a transaction containing at least one constraint conflict, such as an inter-table referential constraint conflict **4** for a cascade-rejection of a transaction dependent on a prior transaction having at least one constraint conflict
IBMSNAP_APPLY_QUAL	This column is blank for local updates and the name of the associated Apply program for foreign updates. The Capture program derives this value from the CRITSEC table.

```
CREATE TABLE ASN.IBMSNAP_UOW (
 IBMSNAP_UOWID CHAR(10) FOR BIT DATA NOT NULL,
 IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA NOT NULL,
 IBMSNAP_LOGMARKER TIMESTAMP NOT NULL,
 IBMSNAP_AUTHTKN CHAR(12) NOT NULL,
 IBMSNAP_AUTHID CHAR(18) NOT NULL)
 IBMSNAP_REJ_CODE CHAR(1) NOT NULL WITH DEFAULT
 IMBSNAP_APPLY_QUAL CHAR(18) NOT NULL WITH DEFAULT
CREATE UNIQUE INDEX ASN.ZIBMSNAP_UOW ON
 ASN.IBMSNAP_UOW (IBMSNAP_COMMITSEQ DESC)
CREATE UNIQUE INDEX ASN.ZIBMSNAP_UOW2 ON
 ASN.IBMSNAP_UOW (IBMSNAP_UOWID DESC)
```

Note: On AS/400 systems, IBMSNAP_AUTHTKN is CHAR(26)

To improve fetch performance, create the following index:

```
CREATE UNIQUE INDEX U1yymmddhhmmssIX1 ON ASN.IBMSNAP_UOW
(IBMSNAP_COMMITSEQ ASC, IBMSNAP_UOWID ASC, IBMSNAP_lOGMARKER ASC)
```

To improve pruning performance, create the following index:

For MVS:

```
CREATE UNIQUE INDEX U2yymmddhhmmssIX2 ON ASN.IBMSNAP_UOW
(IBMSNAP_COMMITSEQ DESC, IBMSNAP_UOWID ASC)
```

For all other platforms:

```
CREATE UNIQUE INDEX U2yymmddhhmmssIX2 ON ASN.IBMSNAP_UOW
(IBMSNAP_
```

Figure 40. CREATE TABLE Statement

End of Product-sensitive programming interface

Diagnosis, Modification or Tuning information

Warm Start Table

- ASN.IBMSNAP_WARM_START

This table contains information that enables the Capture program to resynchronize (see the warm start section of the Capture and Apply programs chapter for your platform). It is created in the same database as ASN.IBMSNAP_REGISTER.

Table 32. Warm Start Table Columns

Column name	**Description**
SEQ	The log RBA for this unit of work. Records Capture's position in the DBMS log. Used for quickly restarting following a shutdown or failure.
AUTHTKN	The DB2 token for the unit of work associated with the SEQ position.
AUTHID	The DB2 authorization ID for the unit of work associated with the SEQ position.
CAPTURED	A flag indicating whether or not this unit of work has been captured. **Y** This unit of work has been captured. **N** This unit of work has not been captured.
UOWTIME	The MVS TOD (time of day) or OS/2 and AIX CUT (coordinated universal time) clock indicating when the unit of work associated with the SEQ position was captured (source server timestamp).

```
CREATE TABLE ASN.IBMSNAP_WARM_START (
 SEQ CHAR(10) FOR BIT DATA,
 UOWID CHAR(10) FOR BIT DATA
 AUTHTKN CHAR(12),
 AUTHID CHAR(18),
 CAPTURED CHAR(1),
 UOWTIME INT);
```

Figure 41. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Diagnosis, Modification or Tuning information

Warm Start Table for Capture for VSE and VM

- ASN.IBMSNAP_WARM_START

This table contains information that enables the Capture program to resynchronize for VSE and VM (see the warm start section for the Capture and Apply programs for VSE and VM). It is created in the same database as ASN.IBMSNAP_REGISTER.

Table 33 (Page 1 of 2). Warm Start for Capture for VSE and VM Table Columns

Column name	**Description**
SEQ	The log RBA for this unit of work. Records Capture's position in the DBMS log. Used for quickly restarting following a shutdown or failure.
UOWID	The unit-of-recovery ID from the log record header for this unit-of-work.

Table 33 (Page 2 of 2). Warm Start for Capture for VSE and VM Table Columns

Column name	Description
AUTHID	The DB2 authorization ID for the unit of work associated with the SEQ position.
CAPTURED	A flag indicating whether or not this unit of work has been captured. **Y** This unit of work has been captured. **N** This unit of work has not been captured.
UOWTIME	The MVS TOD (time of day) or OS/2 and AIX CUT (coordinated universal time) clock indicating when the unit of work associated with the SEQ position was captured (source server timestamp).

```
CREATE TABLE ASN.IBMSNAP_WARM_START (
 SEQ CHAR(10) FOR BIT DATA,
 UOWID CHAR(10) FOR BIT DATA
 AUTHID CHAR(18),
 CAPTURED CHAR(1),
 UOWTIME INT) IN DROPTS;
```

Figure 42. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Control Tables Used at the Control Server

This section, describes tables that are created automatically during the course of defining subscription activities.

Table 34. Control Tables Used at the Control Server

Table name	Internal name	Description
Apply trail table	ASN.IBMSNAP_APPLYTRAIL	This table records a history of updates performed against replication subscriptions. The subscription statistics can be used to audit update activity and is a repository of diagnostics and performance statistics.
Set subscription table	ASN.IBMSNAP_SUBS_SET	This table ensures that each set name is used only once for every Apply qualifier.
Subscription columns table	ASN.IBMSNAP_SUBS_COLS	This table contains information on the common subscription columns being copied in a replication subscription.
Subscription events table	ASN.IBMSNAP_SUBS_EVENT	This table contains information on the events being copied in a replication subscription.
Subscription statements table	ASN.IBMSNAP_SUBS_STMTS	This table contains information on the statements being copied in a replication subscription.
Subscription targets member table	ASN.IBMSNAP_SUBS_MEMBR	This table contains information on the targets member being copied in a replication subscription.

Diagnosis, Modification or Tuning information

Apply Trail Table

- ASN.IBMSNAP_APPLYTRAIL

The Apply trail table records a history of updates performed against subscriptions. The subscription statistics can be used to audit update activity and is a repository of diagnostics and performance statistics.

Table 35 (Page 1 of 3). Apply Trail Table Columns

Column name	Description	
APPLY_QUAL	Is from IBMSNAP_SUBS_SET.	
SET_NAME	Is from IBMSNAP_SUBS_SET.	
WHOS_ON_FIRST	Is from IBMSNAP_SUBS_SET.	
ASNLOAD	**Y or N**	Indicates whether LOAD was called during the subscription set process.
	Null	Indicates an error.
MASS_DELETE	**Y or N**	Indicates whether LOAD was called during the subscription set process.
	Null	if error. If the error occured before the decision to issue a mass delete, then a mass delete will not be executed.
EFFECTIVE_MEMBERS	The number of members associated with calls to ASNLOAD, or the number of members for which rows are fetched and inserted, updated, or deleted.	

Table 35 (Page 2 of 3). Apply Trail Table Columns

Column name	Description
SET_INSERTED	The total number of rows individually inserted into any set members during the subscription cycle.
SET_DELETED	The total number of rows individually deleted from any set members during the subscription cycle.
SET_UPDATED	The total number of rows individually updated in any set members during the subscription cycle.
SET_REWORKED	The total number of inserts reworked as updates and updates reworked as inserts for any set members during the subscription cycle.
SET_REJECTED_TRXS	The total number of rejected replica transactions due to a direct or cascading update conflict. Always zero if either source or target is a replica.
STATUS	From IBMSNAP_SUBS_SET. A value that represents in-progress and completed work status for Apply. **-1** A known failed execution; the definition can be modified. **0** A stable definition row which can be modified. **1** A pending or in-progress execution; this definition should not be modified. **2** A continuing execution of a single logical subscription which was divided according to the MAX_SYNCH_MINUTES control column and is being serviced by multiple subscription cycles.
LASTRUN	The time when this subcycle is supposed to start. This value is calculated by adding the LASTRUN value with the INTERVAL_MINUTES value.
LASTSUCCESS	Is from IBMSNAP_SUB_SET.
SYNCHPOINT	Is from IBMSNAP_SUB_SET.
SYNCHTIME	Is from IBMSNAP_SUB_SET.
SOURCE_SERVER	Is from IBMSNAP_SUBS_SET.
SOURCE_ALIAS	Is from IBMSNAP_SUBS_SET.
SOURCE_OWNER	Is from IBMSNAP_SUBS_SET.
SOURCE_TABLE	Is from IBMSNAP_SUBS_SET.
SOURCE_VIEW_QUAL	Is from IBMSNAP_SUBS_SET.
TARGET_SERVER	Is from IBMSNAP_SUBS_SET.
TARGET_ALIAS	Is from IBMSNAP_SUBS_SET.
TARGET_OWNER	Is from IBMSNAP_SUBS_SET.
TARGET_TABLE	Is from IBMSNAP_SUBS_SET.
SQLSTATE	If an error, the SQL error code. Otherwise, null.
SQLCODE	If an error, the database-specific SQL error code. Otherwise, null.
SQLERRP	If an error, the database product identifier. Otherwise, null.
SQLERRM	If an error, the string containing the SQL error. Otherwise, null.

Table 35 (Page 3 of 3). Apply Trail Table Columns

Column name	Description
APPERRM	The Apply error message text from SQLCA if there was an error, otherwise null. This value is constant for each copy derived, directly or indirectly, from the original source table.

```
CREATE TABLE
ASN.IBMSNAP_APPLYTRAIL (
  APPLY_QUAL CHAR(18) NOT NULL,
  SET_NAME CHAR (18) NOT NULL,
  WHOS_ON_FIRST CHAR(1) NOT NULL,
  ASNLOAD CHAR(1),
  MASS_DELETE CHAR(1),
  EFFECTIVE_MEMBERS INT,
  SET_INSERTED INT NOT NULL,
  SET_DELETED INT NOT NULL,
  SET_UPDATED INT NOT NULL,
  SET_REWORKED INT NOT NULL,
  SET_REJECTED_TRXS INT NOT NULL,
  STATUS SMALLINT NOT NULL,
  LASTRUN TIMESTAMP NOT NULL,
  LASTSUCCESS TIMESTAMP,
  SYNCHPOINT CHAR(10) FOR BIT DATA,
  SYNCHTIME TIMESTAMP,
  SOURCE_SERVER CHAR(18) NOT NULL,
  SOURCE_ALIAS CHAR(8),
  SOURCE_OWNER CHAR (18),
  SOURCE_TABLE CHAR(18),
  SOURCE_VIEW_QUAL SMALLINT,
  TARGET_SERVER CHAR(18) NOT NULL,
  TARGET_ALIAS CHAR(8),
  TARGET_OWNER CHAR (18) NOT NULL,
  TARGET_TABLE CHAR(18) NOT NULL,
  SQLSTATE CHAR(5),
  SQLCODE INTEGER,
  SQLERRP CHAR(8),
  SQLERRM VARCHAR(70),
```

Figure 43. CREATE TABLE Statement

End of Diagnosis, Modification or Tuning information

Product-sensitive programming interface

Subscription Columns Table

- ASN.IBMSNAP_SUBS_COLS

This table contains information on the common subscription columns being copied in a replication subscription. The common subscription columns table contains the replication subscription name, target owner, and target tables where the targets member exists, whether the columns table is part of the primary key or not, and the SQL expression necessary to generate an aggregate column.

A unique index on APPLY_QUAL, SET_NAME, TARGET_OWNER, TARGET_TABLE, and TARGET_NAME is created automatically by the Control Center.

Table 36. Subscription Columns Table Columns

Column name	**Description**
APPLY_QUAL	From IBMSNAP_SUBS_MEMBR.
SET_NAME	From IBMSNAP_SUBS_MEMBR.
WHOS_ON_FIRST	From IBMSNAP_SUBS_MEMBR.
TARGET_OWNER	The middle qualifier of the target table name.
TARGET_TABLE	The third qualifier of the target table name.
COL_TYPE	**A** For after-image column **B** For before-image column **C** For computed column without SQL column function reference. **F** For computed column without SQL column function reference. **R** Signifies a relative record number column, provided by the system and used as a primary key column. Used only by DPropR/400.
TARGET_NAME	The name of the target table column that does not need to match the source column name. **Note:** CCD column names cannot be renamed columns. They must match the CD_TABLE column names.
IS_KEY	**Y** The column is all or part of the primary key of the target (all condensed copies must have primary keys). **N** The column is not part of a key of the target.
COLNO	The numeric location of the column in the original source, to be preserved relative to other user columns in displays and subscriptions.
EXPRESSION	The source column identifier.

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_COLS(
  APPLY_QUAL CHAR (18) NOT NULL,
  SET_NAME CHAR (18) NOT NULL,
  WHOS_ON_FIRST CHAR(1) NOT NULL,
  TARGET_OWNER CHAR(18) NOT NULL,
  TARGET_TABLE CHAR(18) NOT NULL,
  COL_TYPE CHAR(1) NOT NULL,
  TARGET_NAME CHAR(18) NOT NULL,
  IS_KEY CHAR(1) NOT NULL,
  COLNO SMALLINT NOT NULL,
  EXPRESSION VARCHAR(254) NOT NULL);
CREATE UNIQUE INDEX IBMSNAP_SUBS_COLSX ON ASN.IBMSNAP_SUBS_COLS
(APPLY_QUAL, SET-NAME, WHOS_ON_FIRST TARGET_OWNER, TARGET_TABLE,
TARGET_NAME);
```

Figure 44. CREATE TABLE Statement

End of Product-sensitive programming interface

Product-sensitive programming interface

Subscription Events Table

- ASN.IBMSNAP_SUBS_EVENT

This table contains information on the events being copied in a replication subscription. The common subscription events table contains the replication subscription name, target owner, and target tables where the event exists, whether the event is part of the primary key or not, and the SQL expression necessary to generate an aggregate column.

A unique index on EVENT_NAME and EVENT_TIME is created automatically by the Control Center.

Table 37 (Page 1 of 2). Subscription Events Table Columns

Column name	Description
EVENT_NAME	A globally unique character string in a global name space configuration or a control server unique character string.
EVENT_TIME	A control server timestamp of a current or future posting time. User applications signalling replication events provide the values in this column.

Table 37 (Page 2 of 2). Subscription Events Table Columns

Column name	Description
END_OF_PERIOD	A source server timestamp value. An upper bound function that blocks replication until a later event is posted. **Note:** This is the only way to prevent eligible change data from replicating during a subscription cycle. The value in this column must be less than the CURRENT TIMESTAMP value at the source server.

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_EVENT (
  EVENT_NAME CHAR(18) NOT NULL,
  EVENT_TIME TIMESTAMP NOT NULL,
  END_OF_PERIOD TIMESTAMP;
CREATE UNIQUE INDEX IBMSNAP_SUBS_EVENT ON ASN.IBMSNAP_SUBS_EVENT
(EVENT_NAME, EVENT_TIME);
```

Figure 45. CREATE TABLE Statement

End of Product-sensitive programming interface

Product-sensitive programming interface

Subscription Set Table

- ASN.IBMSNAP_SUBS_SET

The subscription set table ensures that each set name is used only once for every Apply qualifier.

Table 38 (Page 1 of 3). Subscription Set Table Columns

Column name	Description
APPLY_QUAL	Identifies the Apply program for the platform instance which will run this subscription.
SET_NAME	From APPLY_QUAL. Names the replication subscription.
WHOS_ON_FIRST	Allows both the up and down replication subscriptions to be contained in the same set without the potential for multiple SOURCE_SERVER values per APPLY_QUAL, SET_NAME pairing.
ACTIVATE	Indicates: **0** The subscription set is deactivated. **1** The request is active indefinitely. **2** The set can be copied immediately.

Table 38 (Page 2 of 3). Subscription Set Table Columns

Column name	Description
SOURCE_SERVER	The RDB name of DB2 for MVS, VSE, and VM where the source tables and views are defined.
SOURCE_ALIAS	The name of the SOURCE_SERVER used by the Apply program on the DB2 Universal Database client interface. This identifies the location of the subscription definition and is later used to perform administrative actions on a subscription. This does not necessarily match the alias of the source server used by the Apply program for the DB2 Universal Database and may be null if the database has no DB2 Universal Database name.
TARGET_SERVER	The RDB name of the MVS, VSE, or VM server where the target table and views are defined.
TARGET_ALIAS	The name of the TARGET_SERVER used by the Apply program on the DB2 Universal Database client interface. This identifies the location of the subscription definition and is later used to perform administrative actions on a subscription. This does not necessarily match the alias of the target server used by the Apply program for the DB2 Universal Database.
STATUS	Records in-progress and completed work status as the Apply program for the platform disconnects from the CNTL_SERVER where the control table is stored to access data at the SOURCE_SERVER and update the copy at the TARGET_SERVER. The completion state cannot be set at the CNTL_SERVER until the copy has been refreshed.
LASTRUN	The estimated time the last subscription style began.
REFRESH_TIMING	Sets the timing between statement executions. **R** Apply uses the value in SLEEP_MINUTES to determine replication timing. **E** Apply checks the time value in the SUBS_EVENT table to determine replication timing. **B** Indicates a subscription having both relative and event timing specifications. In all other instances, use SLEEP_MINUTES.
SLEEP_MINUTES	Specifies the time of inactivity between statement executions.
EVENT_NAME	The optional foreign key into IBMSNAP_SUBS_EVENT in the control server.
LASTSUCCESS	The control server wallclock time of the beginning of a susccessful subscription cycle, recorded at the end of the cycle if STATUS=1 or STATUS=2. If not, the existing value is unchanged.
SYNCHPOINT	The sysnchpoint value of the upper-bound of requested transaction data during the previous subscription cycle. This value may advance even if no change data rows were fetched.
SYNCHTIME	The timestamp associated with the SYNCHPOINT column of the current table.
MAX_SYNCH_MINUTES	A time-threshold limit to regulate the amount of change data to fetch and apply during a subscription cycle. The limit is automatically recalculated if the Apply program encounters a resource constraint making the set limit unfeasible.

Table 38 (Page 3 of 3). Subscription Set Table Columns

Column name	**Description**
AUX_STMTS	A 0 value representing the absence of one or more IBMSNAP_SUBS_STMTS rows for the subscription. The value represents the number of such rows.
ARCH_LEVEL	The architectural level of the definition contained in the row. This field identifies the rules under which a row was created.
BEFORE_OR_AFTER	Specifies: **A** The statement is executed at the TARGET_SERVER and should succeed the application of changes to the target tables. **B** The statement is executed at the TARGET_SERVER and should preceed the application of changes to the target tables. **S** The statement is executed at the SOURCE_SERVER and should preceed the opening of answer set cursors.

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_SET (
    APPLY_QUAL CHAR(18) NOT NULL,
  SET_NAME CHAR (18) NOT NULL,
  WHOS_ON_FIRST CHAR(1) NOT NULL,
  ACTIVATE SMALLINT NOT NULL,
  SOURCE_SERVER CHAR(18) NOT NULL,
  SOURCE_ALIAS CHAR(8),
  TARGET_SERVER CHAR(18) NOT NULL,
  TARGET_ALIAS CHAR(8),
  STATUS SMALLINT NOT NULL,
  LASTRUN TIMESTAMP NOT NULL,
  REFRESH_TIMING CHAR(1) NOT NULL,
  SLEEP_MINUTES INT,
  EVENT_NAME CHAR(18),
  LASTSUCCESS TIMESTAMP,
  SYNCHPOINT CHAR(10) FOR BIT DATA,
  SYNCHTIME TIMESTAMP,
  MAX_SYNCH_MINUTES INT,
  AUX_STMTS SMALLINT NOT NULL,
  ARCH_LEVEL CHAR(4) NOT NULL,
 < + padding where row level locking not supported>)
  This index ensures that each set name is used just once
  per Apply qualifier.  Each Apply process can only have
  one control server, so this index suffices, given that
  the assumption that unique APPLY_QUAL values are used for
  each Apply process.  The set could be cloned, but only
  for a different Apply process using a different
  APPLY_QUAL value.
CREATE UNIQUE INDEX IBMSNAP_SUBS_SETX ON ASN.IBMSNAP_SUBS_SET
(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST);
```

Figure 46. CREATE TABLE Statement

End of Product-sensitive programming interface

Product-sensitive programming interface

Subscription Statements Table

- ASN.IBMSNAP_SUBS_STMTS

This table contains information on the statements being copied in a replication subscription. The common subscription statements table contains the replication subscription name, target owner, and target tables where the statement exists, whether the statement is part of the primary key or not, and the SQL expression necessary to generate an aggregate column.

A unique index on the APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, BEFORE_OR_AFTER, and STMT_NUMBER columns is created automatically by the Control Center.

Table 39. Subscription Statements Table Columns

Column name	**Description**
APPLY_QUAL	From IBMSNAP_SUBS_SET.
SET_NAME	From IBMSNAP_SUBS_SET.
WHOS_ON_FIRST	From IBMSNAP_SUBS_SET.
BEFORE_OR_AFTER	Specifies either: **A** The statement is executed at the target server and should succeed the application of changes to the target tables. **B** The statement is executed at the target server and should succeed the application of changes to the target tables. **S** The statement is executed at the source server and should precede the opening of the answer set cursors.
STMT_NUMBER	Defines the relative order of execution within the scope of BEFORE_OR_AFTER.
EI_OR_CALL	Specifies either: **E** The SQL_STMT should be run as an EXEC SQL EXECUTE IMMEDIATE at the target server. **C** The SQL_STMT contains a stored procedure name to run as an EXEC SQL CALL at the target server.
SQL_STMT	Specifies either: **E** The SQL string runs as an EXEC SQL EXECUTE IMMEDIATE statement. **C** The 8-byte name of an SQL-stored procedure, without parameters or the CALL keyword, runs as an EXEC SQL CALL statement.
ACCEPT_SQLSTATES	One to ten 5-byte SQLSTATE values may be specified as acceptable SQLSTATES. The values would otherwise cause termination of subscription execution.

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_STMTS (
  APPLY_QUAL CHAR (18) NOT NULL,
  SET_NAME CHAR (18) NOT NULL,
  WHOS_ON_FIRST CHAR(1) NOT NULL,
  BEFORE_OR_AFTER CHAR(1) NOT NULL,
  STMT_NUMBER SMALLINT NOT NULL,
  EI_OR_CALL CHAR(1) NOT NULL,
  SQL_STMT VARCHAR(1024),
  ACCEPT_SQLSTATES VARCHAR (50));
CREATE UNIQUE INDEX IBMSNAP_SUBS_STMTX ON ASN.IBMSANP_SUBS_STMTS
(APPLY_QUAL, SET_NAME, WHOS_ON_FIRST, BEFORE_OR_AFTER,
STMT_NUMBER);
```

Figure 47. CREATE TABLE Statement

End of Product-sensitive programming interface

Product-sensitive programming interface

Subscription Targets Member Table

- ASN.IBMSNAP_SUBS_MEMBR

This table contains information on the targets member being copied in a replication subscription. The common subscription targets member table contains the replication subscription name, target owner, and target tables where the targets member exists, whether the targets member is part of the primary key or not, and the SQL expression necessary to generate an aggregate column.

A unique index on APPLY_QUAL, SET_NAME, TARGET_OWNER, TARGET_TABLE, and TARGET_NAME is created automatically by the Control Center.

Table 40 (Page 1 of 2). Subscription Targets Member Table Columns

Column name	Description
APPLY_QUAL	From IBMSNAP_SUBS_SET.
SET_NAME	From IBMSNAP_SUBS_SET. **Note:** For replicas, both the to replica and from replica subscriptions share the same APPLY_QUAL and SET_NAME values determined by their WHOS_ON_FIRST values.
WHOS_ON_FIRST	From IBMSNAP_SUBS_SET.
SOURCE_OWNER	The middle qualifier of the source table name.
SOURCE_TABLE	The third qualifier of the source table name.

Table 40 (Page 2 of 2). Subscription Targets Member Table Columns

Column name	Description
SOURCE_VIEW_QUAL	Supports join subscriptions by matching the similar column in IBMSNAP_REGISTER.
TARGET_OWNER	The middle qualifier of the target name. The value in SOURCE_OWNER should be used as the default.
TARGET_CONDENSED	A flag indicating: **Y** Changes can be netted out, with at most one row in the base table for every original table primary key value. **N** All changes must remain, retaining a complete update history. **A** Valid only for base aggregate or change aggregate tables.
TARGET_COMPLETE	A flag indicating: **Y** The base table contains a row for every primary key value of interest. **N** The base table contains some subset of rows of primary key values.
TARGET_STRUCTURE	**1** source table **3** CCD table **4** point-in-time table **5** base aggregate table **6** change aggregate table **7** replica **8** user copy
PREDICATES	Lists the predicates to be placed in a WHERE clause to subset the view (horizontal fragment) maintained in the TARGET_TABLE. This column can be updated directly with a predicate in situations where an out-of-range insert is performed by a local application.

```
CREATE TABLE
  ASN.IBMSNAP_SUBS_MEMBR (
  APPLY_QUAL CHAR (18) NOT NULL,
  SET_NAME CHAR (18) NOT NULL,
  WHOS_ON_FIRST CHAR(1) NOT NULL,
  SOURCE_OWNER CHAR (18) NOT NULL,
  SOURCE_TABLE CHAR(18) NOT NULL,
  SOURCE_VIEW_QUAL SMALLINT NOT NULL,
  TARGET_OWNER CHAR (18) NOT NULL,
  TARGET_TABLE CHAR(18) NOT NULL,
  TARGET_CONDENSED CHAR(1) NOT NULL,
  TARGET_COMPLETE CHAR(1) NOT NULL,
  TARGET_STRUCTURE SMALLINT NOT NULL,
  PREDICATE VARCHAR (512));
CREATE UNIQUE INDEX IBMSNAP_SUBS_MEMIX ON ASN.IBMSNAP_SUBS_MEMBR (APPLY_QUAL, SET_NAME,
WHOS_ON_FIRST
SOURCE_OWNER, SOURCE_TABLE, SOURCE_VIEW_QUAL, TARGET_OWNER, TARGET_TABLE);
```

Figure 48. CREATE TABLE Statement

Note: You can introduce either source or target view names when there are multiple members with the same fully qualified source and target names to provide the necessary source/target member name uniqueness.

End of Product-sensitive programming interface

Target Tables

Table 41. Target Tables

Table name	Internal name	Description
Base aggregate target table	userid.target_table	This table is a target table that contains data aggregated from a source table.
Change aggregate target table	userid.target_table	This table is a target table that contains data aggregations based on changes from a source table.
Point-in-time target table	userid.target_table	This table indicates a specific commit time from the source server.
Replica	userid.target_table	This table contains a primary key identical to the primary key of the user table.
User copy target table	userid.target_table	This table is identical to a point-in-time table except that the IBMSNAP_LOGMARKER column is not included here.

General-use programming interface

Base Aggregate Target Table

- userid.target_table

Base aggregate tables are target tables that contain data aggregated from a source table.

For base aggregate tables:

- A descending unique index is suggested for IBMSNAP_LLOGMARKER to support query access.
- Before-image user data columns must be nullable.
- Null attributes of the after-image user data columns should match null attributes of the source, except for primary key columns, which should always be NOT NULL.

Table 42. Base Aggregate Target Table Columns

Column name	**Description**
<user columns>	Columns computed from the base table.
IBMSNAP_LLOGMARKER	The current timestamp at the source server at the time of refresh.
IBMSNAP_HLOGMARKER	The current timestamp at the source server at the time of refresh.

```
CREATE TABLE userid.BA_COPY(
 IBMSNAP_LLOGMARKER TIMESTAMP,
 IBMSNAP_HLOGMARKER TIMESTAMP,
 SUM_COL1 CHAR(1),
 SUM_COL3 CHAR(1));
CREATE INDEX BAyymmddhhmmssIX ON ba_copy (IBMSNAP_LLOGMARKER DESC);
```

Figure 49. CREATE TABLE Statement

End of General-use programming interface

General-use programming interface

Change Aggregate Target Table

- userid.target_table

A change aggregate table is a target table that contains data aggregations based on changes from a source table.

For change aggregate tables:

- A non-unique descending index is optional on IBMSNAP_LLOGMARKER.
- Before-image user data columns must be nullable.

- Computed user data columns should be NOT NULL, if your design can determine that the computation will never generate a null value, for example in a concatenation scenario.
- Because CCD tables can contain duplicate rows, the DISTINCT clause should be used when a CCD table is the replication source table of a change aggregate target table.

Table 43. Change Aggregate Target Table Columns

Column name	Description
<User columns, including computed columns>	These columns are computed from change data related to the base table.
IBMSNAP_LLOGMARKER	The oldest (lowest) IBMSNAP_LOGMARKER or (IBMSNAP_LLOGMARKER) value in the CD or CCD table rows being aggregated.
IBMSNAP_HLOGMARKER	The most recent (highest) IBMSNAP_LOGMARKER or (IBMSNAP_LLOGMARKER) value in the CD or CCD table rows being aggregated.

Note: Since CCD tables may contain duplicates, use the DISTINCT clause when sourcing a change aggregate from a CCD table.

```
CREATE TABLE userid.CA_COPY(
  IBMSNAP_LLOGMARKER TIMESTAMP NOT NULL,
  IBMSNAP_HLOGMARKER TIMESTAMP NOT NULL);
  SUM_COL1 CHAR(1),
  SUM_COL3 CHAR(1) ),
CREATE INDEX CAyymmddhhmmssIX ON ca_copy (IBMSNAP_LLOGMARKER DESC);
```

Figure 50. CREATE TABLE Statement

End of General-use programming interface

General-use programming interface

Consistent Change Data Table

- userid.target_table

CCD tables contain committed change data.

The CCD table can optionally exist as:

- A table of data.
- A staging table maintained by one Apply program.

  The result of a join between the CD table and the UOW table can be stored here, so that you perform the join step once only for “fan-out” copying.

- A secondary staging table.
- An external source table for nonrelational and multivendor data.

  This allows IBM Replication to act as a “loading dock” and deliver transaction-consistent data from nonrelational sources.

For CCD tables:

- A non-unique ascending index on IBMSNAP_COMMITSEQ is created. This index improves the performance of updating CCD table copies.
- If the CCD table is condensed (CCD_CONDENSED = Y), a unique index is required for user data primary key columns to maintain the CCD table. CCD copies with TARGET_CONDENSED = N may contain duplicates as a result of reapplied changes following a failure of the Apply program.
- Before-image user data columns must be nullable and therefore cannot be part of a primary key for a condensed CCD table.
- An external CCD table is an alternate source for the original user table. The user table does not include computed columns; therefore, computed columns should not be included in the CCD subscriptions.
- Null attributes of the after-image user data columns should match the null attributes of the source.
- Views of change data tables can be included in view replication sources.
- If an external program, other than the Apply program, maintains the external CCD table, the external program must initialize, maintain, and supply the correct values for the control columns.
- Non-condensed CCD tables can contain duplicate rows as a result of changes reapplied after a failure during the previous copy operation.

Table 44 provides a list and a brief description of each of the CCD table columns.

Table 44. CCD Table Columns

Column name	**Description**
IBMSNAP_INTENTSEQ	Unique identifier for this change; it describes the sequence of a change within a transaction.
IBMSNAP_OPERATION	Character value of I, U, or D, indicating an insert, update, or delete record.
IBMSNAP_COMMITSEQ	The transaction commit sequencing value.
IBMSNAP_LOGMARKER	The approximate commit time at the source server.
<user data columns>	Columns from source tables specified while defining replication sources.

Figure 51 on page 300 shows the index for CCD tables.

```
CREATE TABLE userid.copy_table (
 IBMSNAP_INTENTSEQ CHAR(10) FOR BIT DATA  NOT NULL,
 IBMSNAP_OPERATION CHAR(1) NOT NULL,
 IBMSNAP_COMMITSEQ CHAR(10) FOR BIT DATA  NOT NULL,
 IBMSNAP_LOGMARKER TIMESTAMP NOT NULL,
 KEY1 CHAR(1),
 DATA1 CHAR(1))
CREATE INDEX Zcopy_table ON copy_table(IBMSNAP_COMMITSEQ);
```

Figure 51. CREATE TABLE Statement

End of General-use programming interface

General-use programming interface

Point-in-Time Target Table

- userid.target_table

This table indicates a specific commit time from the source server.

For point-in-time tables:

- A unique index is automatically created and required for the primary key.
- Before-image user data columns must be nullable.
- Null attributes of the after-image user data columns should match null attributes of the source, except for primary key columns, which should always be NOT NULL.

Table 45. Point-in-Time Target Table Columns

Column name	Description
<user key columns>	The primary key of the target table, although it is not necessarily a component of the primary key of the base table. You can use predicates to prevent a null value from being assigned to the key fields of any copies.
<user nonkey columns>	The nonkey data column from the base table.
IBMSNAP_LOGMARKER	The approximate commit time at the source server. This column is null following a full refresh.

```
CREATE TABLE PIT_COPY(
 COL1 CHAR(1) NOT NULL,
 COL3 CHAR(1),
 IBMSNAP_LOGMARKER TIMESTAMP,
 PRIMARY KEY(COL1) );
CREATE UNIQUE INDEX PITyymmddhhmmssIX ON pit_copy (COL1);
```

Figure 52. CREATE TABLE Statement

Note: The primary key requires a unique index.

End of General-use programming interface

General-use programming interface

Replica Target Table

- userid.target_table

This table contains a primary key identical to the primary key of the user table.

Table 46. Replica Target Table Columns

Column name	Description
<user key columns>	The primary key of the target table, although it is not necessarily a component of the primary key of the base table. You can use predicates to prevent a null value from being assigned to the key fields of any copies.
<user nonkey columns>	The nonkey data column from the base table.

Note: The replica must have the same primary key as the source table.

```
CREATE TABLE REPLICA_COPY(
 COL1 CHAR(1) NOT NULL,
 COL3 CHAR(1),
 PRIMARY KEY(COL1) );
CREATE UNIQUE INDEX REPLICAyymmddhhmmssIX ON replica (COL1);
```

Figure 53. CREATE TABLE Statement

End of General-use programming interface

General-use programming interface

User Copy Target Table

- userid.target_table

This table indicates a specific commit time from the source server.

For user copy target tables:

- A unique index is automatically created and required for the primary key.
- Before-image user data columns must be nullable.
- Null attributes of the after-image user data columns should match null attributes of the source, except for primary key columns, which should always be NOT NULL.

Table 47. User Copy Target Table Columns

Column name	**Description**
<user key columns>	The primary key of the target table, although it is not necessarily a component of the primary key of the base table. You can use predicates to prevent a null value from being assigned to the key fields of any copies.
<user nonkey columns>	The nonkey data column from the base table.

```
CREATE TABLE USER_COPY_TARGET(
 COL1 CHAR(1) NOT NULL,
 COL3 CHAR(1),
 PRIMARY KEY(COL1) );
CREATE UNIQUE INDEX USER_COPY_TARGETyymmddhhmmssIX ON user_copy_target (COL1);
```

Figure 54. CREATE TABLE Statement

End of General-use programming interface

Chapter 21. IBM Replication Messages

The following is a list of messages issued by IBM Replication for the Capture and Apply programs and migration. A brief explanation of the status is provided.

IBM Replication messages are intended to aid the user, as well as IBM Development and Support Center personnel.

Unless otherwise stated, all error codes described here are internal error codes used by IBM Service and IBM development. Also, unless otherwise stated, error messages are issued with a return code of 8.

IBM Replication messages are prefixed as follows:

ASN0	The Capture program
ASN1	The Apply program

Besides using the information here, you can obtain explanations for messages by typing the following:

db2 ”<message number>”.

Note: The Replication Administration messages (DBA6001 — DBA6110) are listed in the UDB messages book.

Capture Program Messages

Note: For soft SQL errors, see the DB2 messages manual for your platform.

ASN0001E The Capture program encountered an SQL error.

Parameters:

Routine name is “<name>”
SQL request is “<request>”
table name is “<table_name>”
SQLCODE is “<sqlcode>”
SQLERRML is “<sqlerrml>”
SQLERRMC is “<sqlerrmc>”

Explanation: A nonzero SQLCODE was returned when the Capture program issued an EXEC SQL statement.

User Response: See the messages and codes publication of the DB2 database manager on your platform for information about SQL return codes that use SQLERRML and SQLERRMC as substitution fields. Contact your DBA for more information.

ASN0002E The Capture program could not connect to DB2.

Parameters:

Routine name is “<routine>”
SQLCODE is “<sqlcode>”

Explanation: An error occurred when the Capture program issued either

- a CONNECT function to DB2 for VSE and VM
- a CONNECT function to DB2 Call Attachment Facility (CAF)
- an implicit connect to DB2 for common services

User Response: See DB2 codes in the messages and codes publication of the DB2 database manager on your platform for the appropriate reason code. For DB2 for MVS errors, see the section in the administration guide that describes the Call Attachment Facility. Contact your DBA for questions and diagnosis.

ASN0003E The Capture program could not open the plan.

Parameters:

Routine name is “<routine>”
Return code is “<return_code>”
Reason code is “<reason_code>”
Subsystem is “<subsystem>”
Plan name is “<ASNLPLAN>”

Explanation: An error occurred when the Capture program tried to open the plan, ASNLPLAN.

User Response: See the DB2 Codes section in the messages and codes publication of the DB2 database manager on your platform to find the appropriate reason code. See the appropriate section in the administration guide publication of the DB2 database manager on your platform: “Call Attachment Facility.”

ASN0004E The Capture program could not start the trace.

Parameters:

Routine name is “<routine>”
Return code is “<return_code>”
Reason code is “<reason_code>”

Explanation: An error occurred when the START TRACE DB2 command was issued, or when Capture program read the DB2 log.

User Response: See the DB2 Codes section of in the messages and codes publication of the DB2 database manager on your platform to find the appropriate reason code. For more information, see either of the following sections in the administration guide publication of the DB2 database manager on your platform: “Call Attachment Facility” (CAF) for START TRACE DB2 errors, or the Instrumentation Facility Interface (IFI) for DB2 log read errors, or contact your DBA. If CAF or the IFI returned a message, it is also printed on the system display console.

ASN0005E The Capture program encountered an error while reading the DB2 log.

Parameters:

Routine name is “<routine>”
LSN is “<log_sequence_number>”
Return code is “<return_code>”
Reason code is “<reason_code>”

Explanation: An error occurred when the Capture program read the DB2 log. There might be an SQL error.

For Capture for MVS, a dump has been generated for this message. The output appears in the data set whose name is specified by the CEEDUMP DDNAME on your Capture for MVS invocation JCL.

For IBM DPROPR Capture of the Universal Database, the “<return_code>” value is for the Asynchronous Read Log. For UNIX, the log file might not be in the path.

For Capture for VSE, the “<return code>” is for the VSE/VSAM GET macro.

For Capture for VM, the “<return code>” is for Diagnose X'A4'.

User Response: See the DB2 Codes section in the messages and codes publication of the DB2 database manager on your platform for the appropriate reason code.

For Capture for MVS, see the Instrumentation Facility Interface (IFI) section in the administration guide publication of the DB2 database manager on your platform or contact your DBA.

For Capture for VSE, see the "VSE/VSAM Return and Error Codes" manual for more information.

For VM/ESA, see the VM/ESA Programming Services for more information.

For the IBM DPROPR Capture of the Universal Database, see the active and archived database logs section in the administration guide for common servers or contact your IBM Service Representative.

ASN0006E The Capture program encountered an unexpected log error of unknown log variation. The routine name is “<routine>”.

Explanation: An unexpected log error not reported by either:

- the Instrumentation Facility Interface (IFI) for Capture for MVS, or
- the Asynchronous Read Log API for IBM DPROPR Capture of the Universal Database

occurred while the Capture program was processing the DB2 log records. The Capture program could not determine the type of SQL update associated with the log record.

For Capture for MVS, a dump has been generated for this message. The output appears in the dataset whose name is specified by the CEEDUMP DDNAME on your Capture for MVS invocation JCL.

User Response: Contact your IBM Service Representative.

ASN0007E The Capture program encountered an unexpected log error of unimplemented data type. The routine name is "<routine>".

Explanation: An unexpected log error not reported by either:

- the Instrumentation Facility Interface (IFI) for Capture for MVS, or
- the Asynchronous Read Log API for IBM DPROPR Capture of the Universal Database

occurred while the Capture program was processing the DB2 log records. The Capture program could not determine the type of SQL update associated with the log record.

For Capture for MVS, a dump has been generated for this message. The output appears in the dataset whose name is specified by the CEEDUMP DDNAME on your Capture for MVS invocation JCL.

User Response: Contact your IBM Service representative.

ASN0008I The Capture program was stopped.

Explanation: The IBM Replication administrator stopped the Capture program using one of the valid methods.

Explanation: This message is for your information only.

User Response: No action is required.

ASN0009E The table was created without the DATA CAPTURE CHANGES (DCC) attribute.

Parameters:

Routine name is "<routine>"
Table name is "<table_name>"

Explanation: The source table was defined without the DCC attribute and the Capture program tried to capture changes for the replication source.

User Response:

1. Stop the Capture program.
2. Delete the replication source.
3. Define the replication source again; if you do not have the **Data capture is full-refresh only** check box selected, the Control Center will alter the source table with the DCC attribute.
4. Start the Capture program.

ASN0010E The Capture program cannot obtain enough storage.

Parameters:

Routine name is "<routine>"
Storage required is "<amount>"

Explanation: The Capture program cannot continue processing because not enough free storage is available.

User Response: For Capture for MVS, ensure that the REGION parameter has enough storage allocated to run your job. If necessary, contact your MVS system programmer to determine the method for requesting sufficient storage.

For Capture for VM, a request to obtain virtual storage could not be satisfied. You might need to increase the size of the virtual machine in which Capture program runs.

For Capture for VSE, all available GETVIS storage has been exhausted. You might need to restart the Capture program after allocating a larger partition.

ASN0011E The DB2 compression dictionary is not available or the IFCID 306 buffer is invalid.

Parameters:

Routine code is “<routine_code>”
Reason code is “<reason_code>”

Explanation: In the case of DB2 compression dictionary is not available error, the Capture program attempted to read log records for an old compression dictionary. DB2 for MVS only retains one version of the compression dictionary in memory. DB2 can only decompress log records for a compressed table if the compression dictionary used to compress the log records is still the current compression dictionary.

In the case of the IFCID 306 buffer being invalid, the control information is missing from the buffer.

For both cases, a dump has been generated for this message. The output appears in the dataset whose name is specified by the CEEDUMP DDNAME on your Capture for MVS invocation JCL.

User Response: For the DB2 compression dictionary error, to avoid an unwanted cold start of the Capture program, you must capture all log records for a compressed table before creating a new version of the compression dictionary. Use the KEEPDICTIONARY option to retain the current version of the compression dictionary during routine REORG processing.

When you want a new compression dictionary for the table, you must synchronize running the REORG utility with running your updated applications and the Capture program as follows:

1. Quiesce your updated applications.
2. Let the Capture program capture all logged updates for the compressed table.
3. Use the REORG utility on the compressed table, creating a new compression dictionary.
4. Release your updated applications.

For the IFCID 306 buffer error, ensure all DB2 maintenance is current.

ASN0013E The Capture program required a column that was not defined in the change data (CD) table.

Parameters:

Routine name is “<routine>”
Table name is “<table_name>”

Explanation: The user did not define an IBMSNAP required column in the change data table.

User Response: Ensure that the change data table definition is correct. Refer to Chapter 20, “Table Structures” on page 265 for more information.

ASN0014E The processing of the Capture program has fallen below a minimum level. The log record lags current time by “<number>” seconds. The routine name is “<routine>”.

Explanation: The Capture program terminated because a high DB2 transaction rate caused the Capture program to run slower than the defined minimum level.

User Response: Refer to the Capture and Apply chapter for your platform for more information on the lag limit. Perform a cold start.

ASN0015E The Capture program encountered a storage allocation error.

Parameters:

Routine name is “<routine>”
Storage required is “<amount>”

Explanation: A storage allocation error was detected; sufficient storage is not available. The Capture program might have been installed improperly.

For the Capture program on AIX, you might not have set the soft links for the component files to the shared directory.

User Response: Determine why memory could not be allocated by looking at the operating system and application task status. Contact your system programmer to determine the method of requesting the storage listed in the error message.

For Capture for AIX, determine whether you have set the soft links for the component files.

For Capture for VM, a request to obtain virtual storage could not be satisfied. You might need to increase the size of the virtual machine in which Capture program runs.

For Capture for VSE, all available GETVIS storage has been exhausted. You might need to restart the Capture program after allocating a larger partition.

ASN0016E The Capture program could not begin capturing changes because there was no eligible replication source.

Parameters:

Routine name is “<routine>”
Table name is “<table_name>”

Explanation: The replication source information in the register table has not been defined.

The Capture program started but could not find source tables that were:

- Enabled with the DATA CAPTURE CHANGES option of the CREATE or ALTER TABLE statement.
- Defined as replication sources with the **Data capture is full-refresh only** check box cleared on the Define as Source window.

User Response: Ensure that the register table is defined properly. For more information about the register table, see Chapter 20, “Table Structures” on page 265. Verify that replication sources have been defined.

ASN0017E The Capture program encountered a severe internal error and could not issue the correct error message. The routine name is “<routine>”; the return code is “<return_code>”; the error message number is “<error_message_num>”.

Explanation: The Capture program could not retrieve the message from the Capture program messages file.

User Response: Edit the Capture program error message file. Locate the ASNnnnn error message number to determine which error message should have been issued. See the information about the error message in this listing to determine how to resolve the error.

ASN0018W The Capture program did not process updates made to the register table rows. The routine name is "<routine>"; the table name is "<table_name>".

Explanation: The user changed a replication source definition while the Capture program was running and then issued a REINIT command. The register table, which contains a row for each replication source, might not match the other replication source control tables.

User Response:

1. Stop Capture.
2. Delete the replication source.
3. Redefine the replication source.
4. Start Capture.

ASN0019E The Capture program libraries are not authorized for the Authorized Program Facility (APF).

Explanation: The Capture program cannot process the STOP, SUSPEND, RESUME, or REINIT commands because the STEPLIB libraries are not authorized for APF.

User Response: Authorize the Capture link library for APF.

ASN0020I Netview Generic Alerts Interface failure. The Netview return code is "<return_code>".

Explanation: The Network Major Vector Transport (NMVT) could not be sent to Netview by the program because the program interface failed. This is a secondary informational message.

User Response: See the Netview programming documentation for a description of the return code to determine the interface error. The Capture program alerts will not be received by the System Services Control Point (SSCP) until the error is corrected.

ASN0021I Netview Program to Program Interface unavailable. The Netview return code is "<return_code>".

Explanation: Netview is unavailable. This is a secondary informational message.

User Response: See the Netview programming documentation for a description of the return code to determine the Netview problem. For example, the subsystem might not have been started.

ASN0022E DB2 release "<release>" is not supported. The routine name is "<routine>".

Explanation: The Capture program does not support this release of DB2.

User Response: Run the Capture program with the appropriate release of DB2.

ASN0023I The Capture program successfully reinitialized the register table. The table name is "<table_name>"; the routine name is "<routine_name>".

Explanation: A REINIT command was issued and the updates were successfully made to the Capture program internal control information. This message is for your information only.

User Response: No action is required.

ASN0024I The Capture program did not need to reinitialize the register table. Table “<table_name>” did not change.

Explanation: The REINIT command was issued. No updates were made to the register table since initialization or the last REINIT. This message is for your information only.

User Response: No action is required.

ASN0025I The Capture program reinitialized the register table. Table “<table_name“ has <number>” potentially bad row(s).

Explanation: This message accompanies ASN0018W. Reinitialization was performed as requested despite potential problems reported in ASN0018W.

User Response: See ASN0018W.

ASN0026W The Capture program could not allocate the trace buffer. The routine name is “<routine>”; the storage required is <required_storage>”.

Explanation: A storage allocation error was detected; not enough storage is allocated for the trace buffer. The trace buffer is an information-only feature of the Capture program and the allocated storage is not required for the Capture program to run.

User Response: Contact your system programmer to determine the method of requesting the storage listed in the error message.

For Capture for VM, a request to obtain virtual storage could not be satisfied. You might need to increase the size of the virtual machine in which the Capture program runs.

For Capture for VSE, all available GETVIS storage has been exhausted. You might need to restart the Capture program after allocating a larger partition.

ASN0027W The Capture program is already active.

Explanation: You tried to start more than one the Capture program per DB2 subsystem or database.

For VSE/ESA, Capture for VSE generates a unique lock name for each database. This lock name is already in use, indicating that Capture for VSE is already active for the database.

For VM/ESA, Capture for VM has determined that the resource ID used as a lock is already in use. The resource ID is specified on the ENQ_NAME parameter of the CAPTURE ASNPARMS file.

User Response: For DB2 for MVS subsystems, either run only one instance of the Capture program for all subsystems that are members of a data-sharing group, or run only one instance of the Capture program on any stand-alone system.

For other DB2 database platforms, run only one Capture program per database.

For Capture for VM, you can change the ENQ_NAME parameter in the CAPTURE ASNPARMS to ensure unique values for each Capture program if you want to run Capture for VM for more than one DB2 database on a system.

ASN0028I The Capture program is suspended by operator command.

Explanation: The IBM Replication administrator suspended the Capture program and has entered a wait state. This message is for your information only.

User Response: No action is required.

ASN0029I The Capture program is resumed by operator command.

Explanation: The IBM Replication administrator resumed the Capture program from a suspended state and the Capture program has continued running. This message is for your information only.

User Response: No action is required.

ASN0030I The Capture program command entered by the operator was unrecognized.

Explanation: The IBM Replication administrator entered a command not recognized by the Capture program. The only valid commands are:

STOP (*Ctrl+C* for IBM DPROPR Capture of the Universal Database)
SUSPEND
RESUME
REINIT
PRUNE
GETLSEQ

There are no parameters allowed for these commands.

User Response: Use only valid Capture program commands.

ASN0031E The Capture program tuning parameter table can have only one row. The routine name is “<routine>”; the table name is “<table_name>”.

Explanation: The tuning parameter table was not defined correctly or has been updated with invalid rows.

User Response: Refer to Chapter 20, “Table Structures” on page 265 to determine the correct format of this table. Remove any invalid rows.

ASN0033E The Capture program could not reinitialize the register table. The table name is ”<table_name>“.

Explanation: The IBM Replication administrator tried to reinitialize the Capture program, but there was an error in the register table. A user might have tried to update a replication source while the Capture program was running or suspended, and the register table might not match the other control tables.

User Response: This is a secondary message. See any preceding messages for more information about the error. See the Capture and Apply section for your platform more information about reinitializing the Capture program and Chapter 20, “Table Structures” on page 265 for information about the register table.

ASN0034E An incorrect value was supplied for column “<column>” of the Capture program tuning parameter table. The routine name is “<routine>”; the table name is “<table_name>”.

Explanation: The tuning parameters table does not have the correct values. Values might be out of range.

User Response: Refer to the Capture and Apply section for your platform for more information. Check the lag limit, retention period and commit frequency.

ASN0035W Some rows were found in the register table with an unsupported architectural level. The routine name is "<routine>"; the table name is "<table_name>".

Explanation: The register table version does not match the current version of the Capture program. The current version of the Control Center is not compatible with the version of the Capture program that you are running.

User Response: Refer to Chapter 20, "Table Structures" on page 265 to check the required value for the ARCH_LEVEL column in the register table. Verify that the value in the register table at the source server is correct. If not, use compatible versions of the Control Center and the Capture program.

ASN0036E DB2 was terminated abnormally. The routine name is "<routine>".

Explanation: DB2 was terminated while the Capture program was still active.

For MVS/ESA, VSE/ESA or VM/ESA, DB2 was terminated while Capture program was active and the user did not specify the NOTERM start up operand.

User Response: Start DB2 and start the Capture program.

ASN0037W DB2 was terminated in QUIESCE mode. The routine name is "<routine>".

Explanation: DB2 was terminated while the Capture program was still active.

User Response: Start DB2 and start the Capture program.

ASN0038E The disconnect to DB2 failed. The routine name is "<routine>"; the return code is "<return_code>"; the reason code is "<reason_code>".

Explanation: DB2 was stopped in QUIESCE mode, but user wanted to leave the Capture program running. While terminating the connection to DB2, Capture program received an error returned code from the Call Attachment Facility (CAF).

User Response: Restart Capture program.

ASN0040E An error was returned from the FORK function of "<platform>". The error is "<error_text>".

Explanation: An AIX FORK function returned a negative value. "<Error_text>" describes the error.

User Response: See *AIX Calls and Subroutines Reference'* for information about FORK functions, use the provided error text to determine the error, or contact your IBM Service Representative.

ASN0041E An error was returned while getting the instance name. The reason code is "<reason_code>".

Explanation: The SQLEGINS API of DB2 Universal Database returned an error.

User Response: See the *DB2 for common servers API Reference* for information about the SQLEGINS API to determine the error or contact your IBM Service Representative.

ASN0042E An error was returned from the EXECLP function. The error is "<error_text>".

Explanation: The AIX EXECLP function returned a negative value. "<Error_text>" describes the error.

User Response: See the *AIX Calls and Subroutines Reference'* for information about the EXECLP function or contact your IBM Service Representative.

ASN0043E A child process of ASNLMAIN died.

Explanation: The child process created by ASNLMAIN terminated. Possible causes include:

- A user stopped the child process.
- There is an AIX system problem.

User Response: Check the system processes for conflicts or contact your AIX system programmer.

ASN0044E The child process has not called the dummy process after an extended wait.

Explanation: The child process was unable to call the dummy routine ASNLPVRF. The installation softlinks might not have been set.

User Response: Verify whether the installation softlinks have been set, check the system for problems, or contact your IBM Service Representative.

ASN0045E An error was returned from the MSGRCV function. The error is "<error_text>".

Explanation: The function MSGRCV returned an error. "<Error_text>" describes the error.

User Response: Use the provided error text to determine the error, or contact your IBM Service Representative.

ASN0046E An error was returned from the MSGGET function. The error is "<error_text>".

Explanation: The function MSGGET returned an error. "<Error_text>" describes the error. This error occurs during message handling.

User Response: Use the provided error text to determine the error, or contact your IBM Service Representative.

ASN0047E An error was returned from the FTOK function of "<platform>". The error is "<error_text>".

Explanation: The AIX function FTOK returned an error. "<Error_text>" describes the error.

User Response: See *AIX Calls and Subroutines Reference'* for information about the FTOK function, use the provided error text to determine the error, or contact your IBM Service Representative.

ASN0048E The Capture program could not open the log file. The error is "<error_text>. The error code is "<error_code>".

Explanation: The Capture program could not open the log file. Some possible reasons are:

- The Capture program log file was deleted.
- The user does not have the correct authorization for the Capture program directory.

User Response: Contact your system programmer to determine the error or contact your IBM Service Representative.

ASN0050E The Capture program encountered an error while writing to the error message file.

Explanation: An I/O error occurred while writing to the Capture program log file

User Response: Check the ASN.IBMSNAP_TRACE table for error messages.

ASN0053E An error was returned by the Asynchronous Read Log API (SQLURLOG).

Parameters:

Initial LSN is
"<log_sequence_number>"
FIRSTRead LSN is
"<first_read_LSN>"
lastRead LSN is
"<last_read_LSN>"
CurActive LSN is
"<currently_active_LSN>"
log Recswritten is
"<log_records_written>"
log Byteswritten is
"<log_bytes_written>"

Explanation: The Asynchronous Read Log API returned an SQLCODE in the SQL error message that preceded this message. The information in this message provides additional information about the SQL error.

User Response: See ASN0001E for information about SQLCODEs.

ASN0054E The Capture program did not recognize the invocation parameter.

Explanation: An invalid invocation parameter was entered with the ASNCCP command.

User Response: Enter a valid invocation parameter.

See the Capture and Apply section for you platform for information about valid parameters.

ASN0055E The Capture program encountered an SQLTYPE that is not supported in the origin table.

Parameters:

- Routine Name is "<routine>"
- Column Number is "<column_num>"

Explanation: The Capture program encountered an invalid SQL type. A table might have been defined as a replication source outside the Control Center and contains unsupported SQL types (e.g. LONG or large object types).

User Response: Delete the replication source and use the Control Center to define replication sources to ensure only valid types are defined. Or, when manually defining the replication source, ensure that the table has supported SQL types. See the messages and codes publication of the DB2 database manager on your platform to determine the invalid SQLTYPE.

ASN0056E ASN.IBMSNAP_UOW table does not exist.

Explanation: The unit-of-work (UOW) table might have been dropped, or the source server database might have been dropped.

User Response: Contact your IBM Service representative.

ASN0100I The Capture program initialization is successful.

Explanation: This message is for your information only.

User Response: No action is required.

ASN0101W The Capture program warm start failed because existing data is too old; a cold start will be attempted.

Explanation: The data in the change data tables is older than the value "<current_timestamp_lag_limit>". A cold start will be performed.

User Response: See Chapter 20, "Table Structures" on page 265 for more information about warm and cold starts to determine why Capture program could not warm start.

ASN0102W The Capture program will switch to cold start because the warm start information is insufficient.

Explanation: A problem occurred during the retrieval of the warm start information. The warm start table data was invalid. A cold start will be performed.

For DB2 Universal Database, an Asynchronous Read Log API error occurred while reading the log during warm start. For MVS/ESA, an Instrumentation Facility Information (IFI) error occurred while reading the log during warm start.

User Response: See Chapter 20, "Table Structures" on page 265 for more information about warm and cold starts to determine why Capture program could not warm start.

ASN0103I The Capture program started with: "<server_name>".

Parameters:

- SERVER_NAME is "<server_name>"
- ENQ_NAME is "<enq_name>"
- START_TYPE is "<start_type>"
- TERM_TYPE is "<term_type>"
- PRUNE_TYPE is "<prune_type>"

Explanation: This is an informational message that displays the DB2 server name and the Capture program start up option.

For Capture for VSE and VM, the ENQ_NAME shows the name on which Capture program locks to make sure that there is only one Capture program running for any DB2 database. The lock name can be specified for VM/ESA by changing the ENQ_NAME parameter value in the CAPTURE ASNPARMS file.

User Response: No action is required.

ASN0104I Change capture started for owner “<owner>”, the table name is “<copy_table>” at log sequence number (LSN) “<log_sequence_number>”.

Explanation: The Capture program was started for the table owner and table name at the specified log sequence number (LSN). This message is issued for each origin table for which the Capture program captures changes. This message is for your information only.

User Response: No action is required.

ASN0105I Data that has been copied was pruned from the change data table and the unit-of-work table.

Explanation: This message is for your information only.

User Response: No action is required.

ASN0106I The Capture program is waiting for DB2 to come up.

Explanation: When the Capture program is initially brought up, if DB2 is not up at that time, the Capture program waits until DB2 is up. After DB2 is up, the Capture program makes the connection and begins to capture changes.

If the NOTERM option is specified in the Capture invocation parameters, and DB2 comes down smoothly, the Capture program waits for it to come back up.

User Response: No action is required.

ASN0110E Capture for MVS Storage Dump. The Control Address is “<address>”.

Explanation: This is an informational message printed at the top of storage dumps for severe errors. When a dump is generated for a message, the dump output appears in the dataset whose name is specified by the CEEDUMP DDNAME on your Capture for MVS invocation JCL.

User Response: No action is required.

ASN0115I The warm start control information was not supplied. The routine name is “<routine>”; the table name is “<table_name>”.

Explanation: The warm start table is missing or corrupted. This table provides a faster warm start. The Capture program will warm start.

User Response: No action is required.

ASN0116I The Capture program did not reinitialize the tuning parameters table. The routine name is “<routine>”; the table name is “<table_name>”.

Explanation: The REINIT command was issued, but tuning parameter information from the tuning parameters table was not available. The previous tuning parameter values were retained.

User Response: No action is required.

ASN0117W Warm start control information was not saved. The routine name is “<routine>”; the table name is “<table_name>”.

Explanation: An error occurred that prevented warm start information from being saved in the IBMSNAP_WARM_START table. Warm start can be attempted and can take longer because backup sources will be used.

User Response: No action is required.

ASN0121E The Capture program warm start failed because existing data is too old. The Capture program will terminate.

Explanation: The time of the warm start information exceeded LAG_LIMIT.

User Response: No response required; the Capture program will terminate because WARMNS was specified.

ASN0122E An error occurred while reading the warm start information or DB2 log. The Capture program will terminate.

Explanation: A problem occurred while retrieving the warm start information. The warm start table data was invalid or for MVS, an Instrumentation Facility Interface (IFI) error occurred while reading the log during warm start.

User Response: No response required; the Capture program is terminating because WARMNS was specified.

ASN0123I The highest log sequence number of a successfully captured log record is "<log_sequence_number>".

Explanation: The Capture program saved the highest log sequence number (LSN) in the warm start table. This is the point at which the Capture program finished successfully processing the log data.

User Response: No response required; this message accompanies termination.

ASN0124I The prune command was accepted; the pruning action is queued.

Explanation: The IBM Replication administrator entered the prune command and the Capture program has queued the request. The Capture program will prune the change data (CD) table and the unit-of-work (UOW) table.

User Response: No response required.

ASN0125I The current log sequence number of successfully processed log records is "<log_sequence_number>". The log timestamp is "<timestamp>".

Explanation: Capture program is processing the DB2 log at the log sequence number provided.

User Response: No action is required.

ASN0126E The Capture program encountered a syntax error. The Capture program will terminate.

Explanation: The Capture program encountered the wrong combination of invocation parameters.

User Response: Check the Capture and Apply section for your platform for more information about the START command syntax.

ASN0130I The user requested that the Capture program start reading from the end of the DB2 log.

Explanation: The user specified the WRMSKPM parameter when invoking the Capture program.

User Response: No action is required.

ASN0132I The Capture program was invoked by asncopy with the mobile option.

Explanation: This message is for your information only.

User Response: No action is required.

ASN0133I The Capture program reached the end of the mobile transactions.

Explanation: This message is for your information only.

User Response: No action is required.

ASN0134E The Capture program could not obtain the start of log information when it was invoked by asncopy with the mobile option.

Explanation: The Capture program was unable to locate the point in the log where it needed to start reading information.

User Response: Wait for subsequent messages which will provide more detailed information.

ASN0200E An incorrect parameter "<parameter>" was passed to the Capture program.

Explanation: For VM/ESA, one of the following situations caused an error:

- An incorrect parameter was specified on the ASNCCP invocation command.
- The CAPTURE ASNPARMS file contained an invalid parameter.
- An invalid parameter was specified on the :RESID tag in the RESID NAMES file for the :DBNAME. For example, the RESID could be too long.

For VSE/ESA, an invalid parameter was specified on the ASNCCP invocation command.

User Response: Verify that the parameters supplied are valid. See the Capture and Apply section for your platform for more information about the ASNCCP command.

ASN0201E The Capture program encountered a "<platform>" error. The routine name is "<routine>"; the function name is "<function>"; the return code is "<return_code>".

Explanation: On VM:

- For the LINK function, Capture program encountered an error while attempting to LINK the minidisks identified in the *database* SQLFDEF file.

database is the database identified with the SQLINIT or SQLGLOB commands, the default of SQLDBA.

- For the FSREAD, FSPOINT, or FSTATE function errors, the Capture program encountered an error while trying to read CAPTURE ASNPARMS or the *database* SQLFDEF file.
- For the XCIDRM function, Capture program was unable to obtain the resource ID it uses as a lock to ensure that only one Capture program is active for a DB2 database. The error may have occurred for the following reasons:
  - The virtual machine in which the application is running does not have authority to connect to *IDENT.
  - The virtual machine in which the application is running does not have the authority to declare the resource.

On VSE:

- For the GENCB, MODCB, OPEN, GET, CLOSE, or ENDREQ function errors, Capture program encountered an error while trying to set up or read the database log or directory.
- For the GETVIS, FREEVIS, or XPCC function errors, Capture program encountered an error while trying to perform one of these functions.

User Response: Correct the error as described in the platform documentation. On VM:

- For the LINK function, see *VM/ESA CP Command and Utility Reference* for more information about the return code.
- For the FSREAD, FSPOINT, or FSTATE function errors, see *VM/ESA CMS Application Reference - Assembler*.
- For the XCIDRM function, see *VM/ESA CPI Communications User Guide* for more information the return code.
- For other functions, refer to the platform product application development and command documentation.

On VSE:

- For the GENCB, MODCM, OPEN, GET, CLOSE, or ENDREQ function errors, see *VSE/ESA Messages and Codes Reference* , for more information about the IBM VSE/VSAM macros.
- For the GETVIS, FREEVIS, or XPCC function errors, see *VSE/ESA Systems Macro Reference*.

ASN0202E The USERID parameter was not specified.

Explanation: The USERID parameter is required in the PARM= field on the EXEC job control statement that is passed to the Capture program.

User Response: Add the USERID= parameter, specifying the user ID and password, in the PARM= field and resubmit the job.

ASN0203I Linking to “<diskname>” minidisk“<diskowner>”as “<vdev>”

Explanation: The Capture program is about to issue an internal CP link command to the specified database minidisk.

User Response: If prompted, enter the minidisk password.

Apply Program Messages

ASN1000S An internal error occurred for message number “<number>”. Its substitution fields are “<substitution_field_1>”, “<substitution_field_2>”, “<substitution_field_3>”, “<substitution_field_4>”, “<substitution_field_5>”, “<substitution_field_6>”, and “<substitution_field_7>”. The error code is “<error_code>”. The return code is “<return_code>”.

Explanation: The message file for Apply was installed incorrectly.

User Response: Refer to the installation and configuration information in this book pertaining to your platform. Make sure the message file is installed in the correct directory. If it is, contact your IBM Service representative.

ASN1001E The Apply program encountered an SQL error.

Parameters:

ERRCODE is “<error_code>”
SQLSTATE is “<sqlstate>”
SQLCODE is “<sqlcode>”
SQLERRM is “<sqlerrm>”
SQLERRP is “<sqlerrp>”
server name is “<server_name>”
table name is “<table_name>”

Explanation: An error occurred during the execution of an SQL statement.

User Response: Refer to your database messages reference for SQL.

ASN1002E The critical section table could not be locked.

Parameters:

ERRCODE is “<error_code>”
SQLSTATE is “<sqlstate>”
SQLCODE is “<sqlcode>”
SQLERRM is “<sqlerrm>”
SQLERRP is “<sqlerrp>”
server name is “<server_name>”
table name is “<table_name>”

Explanation: The Apply program could not lock the critical table, probably because of simultaneous activity on the table.

User Response: None required. The Apply program will attempt to get a lock on the critical section table during the next interval or event on the next cycle. If this condition persists, check data base activity for the critical section table.

ASN1003E The Apply program could not connect to the server “<server>”.

Parameters:

error code is “<error_code>”
SQLSTATE is “<sqlstate>”
SQLCODE is “<sqlcode>”
SQLERRM is “<sqlerrm>”
SQLERRP is “<sqlerrp>”

Explanation: The Apply program attempted to connect to the database and received a failing return code because either the database was not up or too may users were accessing it.

User Response: Refer to your database messages reference for SQL.

ASN1010E The Apply program could not insert row “<row>” into the audit trail table due to the following error: “<error_code>”.

Explanation: This is an SQL return code indicating that the audit trail table was not set up with the same structure as the table in Chapter 20, “Table Structures” on page 265.

User Response: Refer to Chapter 20, “Table Structures” on page 265 and your database SQL manual.

ASN1011E The copy request has incompatible source and target attributes. The error code is “<error_code>”.

Explanation: This is an SQL error code indicating that the attributes of the target table must be compatible with the attributes of the source table.

User Response: Refer to the BASE_STRUCTURE column in the register table for the compatibility of the source and target attributes.

ASN1012E The source table structure is invalid. The error code is “<error_code>”.

Explanation: This is an SQL return code indicating that the source table structure in the register table was not set up according to the BASE_STRUCTURE column in the register table.

User Response: Refer to Chapter 20, “Table Structures” on page 265, the BASE_STRUCTURE column in the register table for valid source table structures.

ASN1013E The target table structure is invalid. The error code is “<error_code>”.

Explanation: The target table structure in the subscriptions target member table (ASN.IBMSNAP_SUBS_MEMBR) was not valid.

User Response: Refer to Chapter 20, “Table Structures” on page 265 for valid target table structures.

ASN1014E The Apply program could not find a source for the copy request because it could not find the change data table. The error code is “<error_code>”.

Explanation: The change data table was not defined in the register table because either the Apply program did not find the change data table name in the register table or the source table was not registered correctly.

User Response: Refer to Chapter 20, “Table Structures” on page 265 and verify that the change data table is correctly defined in the register table (ASN.IBMSNAP_REGISTER CD_OWNER, CD_TABLE).

ASN1016I Refresh copying has been disabled. The error code is “<error_code>”.

Explanation: While attempting to perform a full refresh, the Apply program encountered a DISABLE_REFRESH column in the register table which was set on.

User Response: Either turn off the DISABLE_REFRESH column or bypass the Apply program and perform a manual refresh.

ASN1017E Apply could not find any target column names. The error code is “<error_code>”.

Explanation: Apply could not find any columns in the ASN.IBMSNAP_SUBS_COLS subscription columns table.

User Response: Refer to “Defining Replication Subscriptions” to redefine the replication subscription.

ASN1019E The target table does not have any key columns. The error code is “<error_code>”.

Explanation: The Apply program could not find key column names in one of the columns requiring a primary key.

User Response: Refer to “<Defining Replication Descriptions>” to redefine the replication description.

ASN1020S The Apply program could not reserve a storage block. The error code is “<error_code>”.

Explanation: The Apply program could not obtain the required (memory) storage.

User Response: Contact your IBM Service representative.

ASN1021S The Apply program could not read the spill file. The error code is “<error_code>”.

Explanation: Either the user does not have the proper access authority for one or all of the files or not enough space is left after writing to the target file.

User Response: Determine whether the problem is caused by lack of access authority or lack of space and contact your system administrator to obtain what is needed.

ASN1022S The Apply program could not write into the spill file. The error code is “<error_code>”.

Explanation: Either the user does not have the proper access authority for one or all of the files or not enough space is left after writing to the target file.

User Response: Determine whether the problem is caused by a lack of access authority or a lack of space and contact your system administrator to obtain what is needed.

ASN1023S The Apply program could not open the spill file. The error code is “<error_code>”.

Explanation: The Apply program could not open the spill file.

User Response: Contact your IBM Service representative.

ASN1024S The Apply program could not close the spill file. The error code is “<error_code>”.

Explanation: The Apply program could not close the spill file.

User Response: Contact your IBM Service representative.

ASN1029E The SQL statement could not execute. The error code is “<error_code>”.

Explanation: The execution of the SQL statement specified by the user was not successful.

User Response: Refer to the SQLSTATE, SQLCODE, SQLERRO, and SQLERRM in the audit trail table and your database SQL manual for detailed information.

ASN1030S The Apply program encountered an OS/2 error. The error code is “<error_code>”; the return code is “<return_code>”.

Explanation: The execution of an OS/2 API failed.

User Response: For more information on the return code, refer to the *OS/2 WARP Control Program Programming Reference*.

ASN1031E The SQL statement is empty. The error code is <error_code>.

Explanation: The SQL statement is an empty string.

User Response: Specify the SQL statement to be executed.

ASN1032S The Apply program log file could not be opened. The error code is “<error_code>”; the return code is “<return_code>”.

Explanation: The Apply program could not open the log file.

User Response: For more information on the return code, either refer to the *OS/2 WARP Control Program Programming Reference* or to the system library information for your particular platform.

ASN1033E The Apply program could not write to the Apply log file. The error code is “<error_code>”; the return code is “<return_code>”.

Explanation: The Apply program could not write to the log file.

User Response: For more information on the return code, either refer to the *OS/2 WARP Control Program Programming Reference* or to the system library information for your particular platform.

ASN1034E Stored procedures are not supported in DB2 for MVS/ESA V3. The error code is “<error_code>”.

Explanation: DB2, Version 3 does not support the stored procedure call.

User Response: Remove the stored procedure CALL statement from the statement table (ASN.IBMSNAP_SUBS_STMT).

ASN1035E The Apply program could not access the subscription columns table.

Parameters:

error code is “<error_code>”
SQLSTATE is “<sqlstate>”
SQLCODE is “<sqlcode>”
SQLERRM is “<sqlerrm>”
SQLERRP is “<sqlerrp>”
server name is “<server_name>”
table name is “<table_name>”

Explanation: An error occurred during the execution of an SQL statement.

User Response: Refer to your database messages reference for SQL.

ASN1036E The column type “<col_type>” for expression “<expression>” is invalid. The error code is “<error_code>”.

Explanation: The value for the COL_TYPE column in the subscription columns table is invalid.

User Response: Change the value to A, B, C, F, or R.

ASN1037E The Apply program could not obtain the date and time. The error code is “<error_code>”; the return code is “<return_code>”.

Explanation: The OS/2 API DosGetDateTime failed.

User Response: For more information on the return code, refer to the *OS/2 WARP Control Program Programming Reference*.

ASN1038E No column names or expressions were specified in the subscription columns table.

Explanation: Column names or expressions for a copy statement must be specified.

User Response: Refer to the "Defining Replication Sources" section of the "Administering Your Replication System" chapter for more information about requirements for subscription definitions.

ASN1039S The Apply program plan, “<plan_name>”, could not be opened.

Parameters:

error code is “<error_code>”
return code is “<return_code>”
reason code is “<reason_code>”

Explanation: The Apply program plan could not be opened.

User Response: Refer to the "Apply for MVS Program Directory; make sure the plan name in the BIND JCL is ASNAP210.

ASN1040S The Apply program encountered an MVS error. The error code is “<error_code>”; the return code is “<return_code>”.

Explanation: Execution of an MVS system operation failed.

User Response: Refer to your MVS system library information.

ASN1041I The Apply program was started using subsystem name: “<subsystem>”.

Explanation: This is not an error message, however, you should make sure that the displayed subsystem name is valid.

User Response: Verify that the subsystem name is valid.

ASN1042W There are too many invocation parameters.

Explanation: The number of parameters you specified when you invoked theApply program exceeds the maximum allowed.

User Response: Refer to the Capture and Apply section for your platform for information on the appropriate number of invocation parameters.

ASN1043E There is already one Apply instance running with this Apply program qualifier “<qualifier>”. The error code is “<error_code>”; the reason code is “<reason_code>”.

Explanation: Verification attempt failed.

User Response: Make sure that only one instance of the Apply program is running under this user ID on this subsystem or database.

ASN1044I The Apply program will become inactive for “<number>” minutes and “<number>” seconds.

Explanation: This message is for your information only.

User Response: No action is required.

ASN1045I The Apply program was started using database “<database>”.

Explanation: This message is for your information only.

User Response: No action is required unless this is not the intended database.

ASN1046S The Apply program libraries are not authorized for the Authorized Program Facility (APF).

Explanation: The Apply program libraries must be APF authorized.

User Response: Authorize the Apply libraries.

ASN1048E The execution of a copy statement failed. See the Apply trail table for full details: ““<text>”

Explanation: A copy statement could not execute. In the message, “<text>” identifies the “<copy_server>”, “<copy_owner, copy_table, stmt_number>”, and “<cntl_server>”.

User Response: Check the APPERRM fields in the audit trail table to determine why the copy statement failed.

ASN1049S The Apply program encountered a system error. The error code is “<error_code>”. The return code is “<return_code>”.

Explanation: Execution of a system operation failed.

User Response: Refer to the system library information for your particular platform.

ASN1050E The Apply program encountered an invalid operation while updating the target table. The error code is “<error_code>”. The invalid operation to be applied is “<operation>”.

Explanation: The operation field of a row fetched from the source table is not valid.

User Response: Contact your IBM Service Representative.

ASN1051E The Apply program detected a gap between the source "<source>" table and the target table. The error code is "<error_code>".

Explanation: The Apply program has detected that the Capture program had lost change data before the Apply program could copy it. For example, the Capture program may have been cold started.

User Response: Check the control tables to determine why the gap is present. Take proper action to preserve data integrity before you reset the control table information to execute the definition again.

ASN1052E The Apply program could not find the ASNLOAD program.

Explanation: The Apply program cannot find the ASNLOAD program in the current directory.

User Response: Make sure that ASNLOAD is in the directory from which you are invoking the Apply program.

ASN1053E The execution of the ASNLOAD program failed. The return code is "<return_code>".

Explanation: The ASNLOAD program detected an error.

User Response: Refer to the messages files generated by the EXPORT and IMPORT utilities. Note that these files names are different for Apply for OS/2 and Apply for AIX.

ASN1056E The Apply program password file could not be opened. The error code is "<error_code>".

Explanation: The user-created password file does not exist.

User Response: If you want to use the AUTHENTICATION=SERVER scheme, you must provide a password file as described in the Apply program section in the Capture and Apply chapter for your platform.

ASN1057E The Apply program could not read the password in the Apply password file. The error code is "<error_code>".

Explanation: The Apply program found no password.

User Response: If you want to use the AUTHENTICATION=SERVER scheme, you must provide a password as described in the Apply program section in the Capture and Apply chapter for your platform.

ASN1058E The Apply program could not close the password file. The error code is "<error_code>".

Explanation: The Apply program could not close the password file.

User Response: Contact your IBM Service representative.

ASN1059E The Apply program detected invalid syntax for line "<line>" in the password file. The error code is "<error_code>".

Explanation: The Apply program could not recognize a line in the password file.

User Response: Correct the syntax error in the password file. See the Apply program section in the Capture and Apply chapter for your platform.

ASN1060E The dynamic allocation for the temporary work file failed. The error code is “<error_code>”.

Explanation: A system error was encountered during dynamic allocation.

User Response: Contact your IBM Service representative.

ASN1061E An invalid keyword parameter was specified. The error code is “<error_code>”.

Explanation: An invalid invocation parameter was specified and ignored by the Apply program.

User Response: Correct the invocation parameter. See the Apply program section in the Capture and Apply chapter for your platform.

ASN1063E A subscription set cannot have more than 200 members. The error code is “<error_code>”.

Explanation: The number of subscriptions has exceeded the maximum allowed number of 200.

User Response: Remove excess members from the subscription.

ASN1065E The Apply program detected a referential integrity violation for UOWID “<ID_number>”. The error code is “<error_code>”.

Explanation: The Apply program has detected an error in the source data.

User Response: Correct the referential integrity problem and rerun the Apply program.

ASN1066S An internal Apply program error occurred. The error code is “<error_code>”.

Explanation: An internal Apply error occurred.

User Response: Contact your IBM Service representative.

ASN1067E The Apply program has detected update conflicts and compensated rejected transactions. See the unit-of-work table for details. The error code is “<error_code>”.

Explanation: More than one application updated the same row in a table from different locations. Some transactions have been rejected and compensated.

User Response: See the ASN.IBMSNAP_UOW table for details.

ASN1068E The Apply program has deactivated the subscription due to a RI violation. The error code is “<error_code>”.

Explanation: A referential integrity violation was detected when copying data from the source table to a replica. The Apply program has terminated and the subscription has been deactivated.

User Response: Correct the referential integrity error and reactivate the subscription.

ASN1069E The Apply program has deactivated the subscription due to a RI violation. All the affected units-of-works have been marked in the unit-of-work table and compensated. The error code is “<error_code>”.

Explanation: A referential integrity violation was detected when replicating data from the replica to the user table. The Apply program has terminated and the subscription has been deactivated.

User Response: Correct the referential integrity error and reactivate the subscription.

ASN1070E The Apply program could not lock the target table.

Parameters:

ERRCODE is "<error_code>"
SQLSTATE is "<sqlstate>"
SQLCODE is "<sqlcode>"
SQLERRM is "<sqlerrm>"
SQLERRP is "<sqlerrp>"
server name is "<server_name>"
table name is "<table_name>"

Explanation: The Apply program could not lock the target tables before it was to check update conflicts.

User Response: Verify that all the target tables are available before rerunning Apply.

ASN1071E The Apply program has detected an error while reading the temporary work file. The error code is "<error_code>".

Explanation: The Apply program has detected an error while reading the temporary work file.

User Response: Contact your IBM Service representative.

ASN1072E The Apply program could not find the ASNDONE program.

Explanation: The Apply program could not find the user exit program, ASNDONE.

User Response: Verify that the ASNDONE program is located in the correct directory.

ASN1073E The execution of the ASNDONE program failed. The return code is "<return_code>".

Explanation: An error occurred while calling the user exit, ASNDONE.

User Response: Contact your IBM Service representative.

ASN1097I The Apply program stopped due to the above error.

Explanation: The error reported previously caused the Apply program to stop.

User Response: Fix the error reported before this message.

ASN1100I A user has stopped the Apply program.

Explanation: A user issued the STOP command to stop the Apply program.

User Response: No action is required.

ASN1201S Place holder for generic message - internal error

Explanation: The asncopy program encountered an SQL error.

Parameters:

ERRCODE is "<error_code>"
SQLSTATE is "<sqlstate>"
SQLCODE is "<sqlcode>"
SQLERRM is "<sqlerrm>"
SQLERRP is "<sqlerrp>"
server name is "<server_name>"

table name is “<table_name>”

User Response: Refer to your database messages reference for SQL.

ASN1202E The asncopy program encountered an SQL error. ERRCODE is “<error code>”, SQLSTATE is “<sqlstate>”, SQLCODE is “<sqlcode>”, SQLERRM is “<sqlerrm>”, SQLERRP is “<sqlerrp>”, table name is “<table name>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN1203I The asncopy program was unable to resume the previous request after being stopped by the user.

Explanation: This message is for your information only.

User Response: No action is required.

ASN1204E The asncopy program encountered an incorrect keyword. The keyword is “<keyword>”.

Explanation: A keyword was entered incorrectly.

User Response: Execute the command again, using the correct keyword.

ASN1205E The asncopy program terminated due to a Capture program error.

Explanation: An inconsistency in Capture program executions has caused the asncopy program to end.

User Response: Refer to the trace produced by the Capture program (ASN.IBMSNAP_TRACE) or the asncopy program error log to determine the cause of the error.

ASN1206E The asncopy program terminated due to an Apply program error.

Explanation: An inconsistency in Apply program executions has caused the asncopy program to end.

User Response: Refer to the apply trail table or the asncopy program error log to determine the cause of the error.

ASN1207E The subscription for “<subscription>” was not enabled. The error code is “<error code>”.

Explanation: The selcted subscription is inactive.

User Response: Either activate the subscription or select another one.

ASN1208E The asncopy program could not find the subscription definition for “<set>”.

Explanation: The selected subscription does not exist.

User Response: Enter the correct subscription.

ASN1209E The asncopy program could not find any eligible subscriptions. The error code is “<error code>”.

Explanation: Either no subscription name was specified or the names specified are invalid.

User Response: Check the subscription names and be sure to enter the correct ones.

ASN1210E The asncopy program was unable to copy subscription definitions.

Explanation: You must specify an Apply qualifier following the keyword q.

User Response: Specify an Apply qualifier following the keyword q.

ASN1211E The asncopy program encountered a timeout. The error code is “<error code>”.

Explanation: You must specify the set names following the keyword (O, U, D, or S).

User Response: Reinitiate the asncopy program, specifying the keyword and then the set names.

ASN1212E The asncopy program encountered a read-only set name “<set_name>” following the keyword “<keyword>”.

Explanation: A read-only set name was specified following the keyword U or D.

User Response: Specify only replica for the keywords U and D.

ASN1214E The set name “<set_name>” is specified more than once.

Explanation: The same set name cannot be specified in more than one list.

User Response: Reinitiate the asncopy program, being sure to specify each set name only once for each list.

ASN1221I Set “<set_name>” has been successfully refreshed with “<>” rows at “<>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN1222S The asncopy program could not write into the file “<set_name>”. The error code is “<error_code>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN1223E The Apply program could not copy for set “<file_name>”. The error code was “<error_code>”.

Explanation: The Apply program encountered a problem while copying.

User Response: Refer to the apply trail table or the asncopy program error log to determine the cause of the error.

ASN1230S The asncopy program encountered a system error. The error code is “<error_code>” and the return code is “<return_code>”.

Explanation: The asncopy program encountered an error in the database.

User Response: Trace the error and call your IBM Service representative.

Migration Messages

ASN2700E The server name “<server_name>” is invalid.

Explanation: The first character of the database server name is invalid.

User Response: Refer to the section on naming conventions for information on valid server names. Then provide the correct server name.

ASN2701E The database “<database_name>” or subsystem “<subsystem_name>” must be an 8-bytes-or-less alias name.

Explanation: The server name consists of more than eight characters.

User Response: Enter the correct alias name. It must consist of eight characters or less.

ASN2702I SQL CONNECT to“<server_type>” “<server_name>” succeeded.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2703E SQL CONNECT to“<server_type>” “<server_name>” failed. Sqlstate is “<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The migration tool cannot connect to the server.

User Response: Refer to the SQLSTATE values section of your database messages and codes manual.

ASN2704I Action “<action>” for “<table_name>” table starts.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2705I Action “<action>” for “<view_name>” view starts.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2706I Action “<action>” for “<table_name>” table succeeded.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2707I Action “<action>” for “<view_name>” view succeeded.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2708E Action "<action>" for "<table_name>" table failed.

Explanation: The migration fallback or the cleanup for the table failed.

User Response: Refer to the previous SQL statement execution to find the SQLSTATE value for the error message.

ASN2709E Action "<action>" for "<view_name>" view failed.

Explanation: The action for the view, MIGRATE, FALLBACK or CLEANUP, failed.

User Response: Refer to the previous SQL statement execution for the SQLSTATE value for the error message.

ASN2710E The BUILDDB action was not done for subscriber "<subscriber_userid>" at "<server_type>" "<server_name>".

Explanation: The BUILDDB step was not executed for the subscriber.

User Response: Perform the BUILDDB and the PREPARE step for the subscriber.

ASN2711E The BUILDDB action was not done for "<server_type>" "<server_name>".

Explanation: The BUILDDB step was not executed for the server.

User Response: Perform the BUILDDB and PREPARE steps for the server.

ASN2712E The requested version "<version>" is invalid.

Explanation: The control table version value entered is not valid.

User Response: Correct the value and reenter it or enter L to run the latest version.

ASN2713E The PREPARE action must be executed before the MIGRATE or FALLBACK action.

Explanation: The PREPARE step was not executed before the MIGRATE step.

User Response: Perform the PREPARE step before performing the MIGRATE step.

ASN2714I Action "<action>" for "<server_type>" "<server_name>" started or action "<action>" for "<subscriber>""<userid>".

Explanation: This message is for your information only.

User Response: No action is required.

ASN2715I Action "<action>" for "<server_type>" "<server_name>" succeeded or action "<action>" for "<subscriber>""<userid>".

Explanation: This message is for your information only.

User Response: No response is required.

ASN2716E Action “<action>” for “<server_type>” “<server_name>” failed or action “<action>” for “<subscriber>”“<userid>”.

Explanation: The migration, fallback, or cleanup for the server failed.

User Response: Refer to the previous SQL statement execution for the SQLSTATE value and then refer to the listing of that value in the message information for your platform to determine the cause of the error.

ASN2717E Data server “<data_server_name>” has not been migrated yet.

Explanation: The data server with dependent subscriptions has not been migrated yet.

User Response: Migrate the data server.

ASN2718I Data server “<data_server_name>” has been migrated.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2719E Action “<action>” for table “<table_name>” was not performed.

Explanation: The action for the table was not executed.

User Response: Repeat the action.

ASN2720E Action “<action>” for view “<view_name>” was not performed.

Explanation: The action for the view was not executed.

User Response: Repeat the action.

ASN2721I Current userid is “<user_id>” and “<server_type>” alias is “<alias_name>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2722I No registration information was collected at version “<version>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2723I No subscription information was collected at version “<version>”.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2724I Table “<table_name>” already exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2725I View “<view_name>” already exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2726I Table “<table_name>” does not exist.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2727I View “<view_name>” does not exist.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2728I More than one table “<table_name>” exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2729E SQL message is “<sqlmsg>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2730E No alias name was found for the server “server_type>” “server_name>”.

Explanation: No alias name was found for the “server_type>” “server_name>” was not provided at the BUILDDB step.

User Response: Provide the complete list of server alias names to the BUILDDB program and run the program again.

ASN2733I The global control information of version “<version>” has been prepared at “<”timestamp>.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQLSTATE values section of your database messages and codes manual.

ASN2734E The file “<file_name>” cannot be opened.

Explanation: This file is either in error or does not exist.

User Response: Verify that the file has been installed correctly. If it has but it still cannot be opened, contact your IBM Service representative. If it has not been installed correctly, reinstall it.

ASN2735E The requested version “<version>” was not found.

Explanation: The information for the requested version cannot be found.

User Response: See the ASN.VERSION_CONTROL table for the available version number specified.

ASN2736E The action parameter was not provided, or was not specified before the keyword parameters.

Explanation: The value for one of the actions (MIGRATE, FALLBACK, or CLEANUP) was not entered before the keyword parameters.

User Response: Enter the action parameter before the keyword parameters.

ASN2737E Too many parameters were entered.

Explanation: The number of parameters entered exceeded the number allowed for the action.

User Response: Verify the necessary parameters and reenter the command.

ASN2738E This parameter is not a keyword parameter: “<parameter_name>”.

Explanation: The parameter entered is not a keyword parameter.

User Response: Enter ASNMIG to see the help for the command syntax. Then enter the correct keyword parameter.

ASN2739E Invalid action parameter was entered. Valid actions are BUILDDB, PREPARE, MIGRATE, FALLBACK, and CLEANUP.

Explanation: The action parameter entered is not a valid value.

User Response: Enter ASNMIG to see a listing of the valid action parameters. Then enter the correct action parameter.

ASN2740E Invalid parameter: “<keyword_parameter>”.

Explanation: The parameter segment entered was not one of the following: D=, S=, C=, T=, U=, or V=.

User Response: Enter the correct parameter.

ASN2741E Invalid combination of keyword parameters.

Explanation: The keyword parameters specified for the action were incorrectly entered.

User Response: See "Migrating DPROPR from Version 1 to Version 5" for the keywords needed for migration.

ASN2742E SQL SELECT failed. Sqlstate is “<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2743I The column being added already exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2744I The row being inserted already exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2745I The row being updated already exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2746I The V5 registrations defined after the last migration were lost.

Explanation: The Version 5 registrations defined after the last migration from Version 1 to Version 5 were lost because of fallback.

User Response: Register Version 5 after the next migration.

ASN2747I The action "<action>" ended successfully.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2748E The action "<action>" ended unsuccessfully.

Explanation: An error occurred during the BUILDDB or PREPARE action.

User Response: Find the error in the action trace file and rerun the action.

ASN2749E The "<action>" partially succeeded because it could not connect to some servers.

Explanation: A wrong server name was provided.

User Response: Rerun the action using the correct server name.

ASN2750E No V1 control server name exists in "<userid>" ROUTING1 table. Fallback from V5 to V1 cannot be performed.

Explanation: The subscriber's routing table contains no rows so fallback cannot be performed.

User Response: Determine why the ROUTING1 table is empty and rebuild it.

ASN2751E SQL PREPARE failed. Sqlstate is"<sqlstate>", sqlcode is "<sqlcode>", sqlerrm is "<sqlerrm>", sqlerrp is "<sqlerrp>".

Explanation: The subscriber's routing table contains no rows so fallback cannot be performed.

User Response: Determine why the routing table is empty and rebuild it.

ASN2752E SQL OPEN CURSOR failed. Sqlstate is“<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2753E SQL FETCH CURSOR failed. Sqlstate is“<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2754I SQL CLOSE CURSOR failed. Sqlstate is“<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2755I This PREPARE run is for validation only.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2756I Action “<action>” trace file is “<action>”.TRC file.

Explanation: The program trace containing records of the action is in the “<action>”.TRC file BUILDDB, PREPARE, MIGRATE, FALLBACK, or CLEANUP).

User Response: Check the trace file to see if there were any errors.

ASN2757I Action “<action>” report file is “<action>”.RPT file

Explanation: This file is created during the migration (PREPARE, MIGRATE, FALLBACK, or CLEANUP).

User Response: Check the report file to see the summary of migration activity and resolve any errors before continuing with migration.

ASN2758E The migration tool cannot execute on this platform “<platform_name>”.

Explanation: Migration could not be performed because of incompatibility between the migration tool and the database it is running on. The problem is probably caused by an error in the installation of the program.

User Response: Refer to the installation section and reinstall the program. Then run the migration tool.

ASN2759E Invalid optional parameter “<parameter_value>”. Accepted parameters are 'V' for Validate and 'O' for Override.

Explanation: The only acceptable validation parameter is V.

User Response: Specify V for the validation parameter and reenter it.

ASN2760I Checking if auto-registration for subscriber “<userid>” exists.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2761W Auto-registration has been detected at the copy server “<server_name>”. The ASN.IBMSNAP_CD_CNTL and pruning control tables need to be migrated first.

Explanation: The ASN.IBMSNAP_CD_CNTL and pruning control tables may not have been migrated at the copy server.

User Response: Migrate the ASN.IBMSNAP_CD_CNTL and pruning control tables at the copy server.

ASN2762I Please issue the following command: ASNMIG “<action>” D=“<migration db>” S=“<V1 data server>” V="<version>".

Explanation: This message follows ASN2761W. The ASN.IBMSNAP_CD_CNTL and pruning control tables may not have been migrated at the copy server.

User Response: Issue the command recommended in the message.

ASN2763I No auto-registration for subscriber “<userid>” has been detected.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2764I The ASN.IBMSNAP_CD_CNTL and pruning control tables created from auto-registration task for subscriber“<subscriber>”'s subscription have been migrated.

Explanation: This message is for your information only.

User Response: No action is required.

ASN2765I The action “<action>” SQL file is “<action>”.SQL file.

Explanation: The SQL statement generated by the BUILDDB action is written to the “<action>”.SQL file (a DB2 statement file).

User Response: Use the DB2 Command Line Processing (CLP) facility to execute the SQL statement in the file.

ASN2766W Auto-registration has been detected at the target server “<server_name>”. The ASN.IBMSNAP_REGISTER and ASN.IBMSNAP_PRUNCNTL tables can be fallen back, but not necessary.

Explanation: The ASN.IBMSNAP_REGISTER and the ASN.IBMSNAP_PRUNCNTL tables may not have fallen back at the copy server.

User Response: Fall back the ASN.IBMSNAP_REGISTER and the pruning control tables at the copy server, if desired.

ASN2767E The list of servers must include at least one valid database alias preceding the wildcard character (*).

Explanation: Either all the server aliases entered were invalid or no alias was entered with the wildcard character (*). The wildcard character alone is not sufficient for the program to run.

User Response: Reenter the wildcard character with the proper server alias.

ASN2768W The wildcard character (*) was not specified, so ensure that you did include the complete list of servers for collection.

Explanation: No wildcard character (*) was entered during the BUILDDB action.

User Response: You can continue with the provided server list if this is your intention. If not, reenter the server list with the wildcard character.

ASN2769E SQL EXECUTE/EXECUTE IMMEDIATE failed. Sqlstate is “<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2770E All Applys having subscriptions from data server “<server_name>” must fall back first.

Explanation: At least one Apply subscribed from this data server has not fallen back yet.

User Response: Perform the FALLBACK for all Applys before the Capture.

ASN2771I Subscriber “<userid>” does not have rows in the subscription tables.

Explanation: The IBM Replication V5 tables contain no information that would allow a fallback to DPROPR V1 for that subscriber.

User Response: Verify that the subscriber userid is correct. If it is, review the executed actions.

ASN2772E SQL COMMIT failed. Sqlstate is “<sqlstate>”, sqlcode is “<sqlcode>”, sqlerrm is “<sqlerrm>”, sqlerrp is “<sqlerrp>”.

Explanation: The SQL statement execution failed.

User Response: Refer to the SQL message and the SQLCODE values section of your database messages and codes manual.

ASN2773I To change to override option, reissue the command with the parameter: "O".

Explanation: The PREPARE action has been performed at the specified version.

User Response: To restart the program, specify the additional "O" to override the values in the DPROP V1 global control table in the migration database.

Part 7. Appendixes

Appendix A. How the DB2 Library Is Structured

The Database Server library consists of SmartGuides, online help, and books. This section describes the information that is provided, and how to access it.

To help you access product information online, DB2 provides the Information Center on OS/2, Windows 95, and the Windows NT operating systems. You can view task information, DB2 books, troubleshooting information, sample programs, and DB2 information on the Web. “About the Information Center” on page 350 has more details.

SmartGuides

SmartGuides help you complete some administration tasks by taking you through each task one step at a time. SmartGuides are available on OS/2, Windows 95, and the Windows NT operating systems. The following table lists the SmartGuides.

SmartGuide	Helps you to...	How to Access...
Add Database	Catalog a database on a client workstation.	From the Client Configuration Assistant, click on **Add**.
Create Database	Create a database, and to perform some basic configuration tasks.	From the Control Center, click with the right mouse button on the **Databases** icon and select **Create**->**New**.
Performance Configuration	Tune the performance of a database by updating configuration parameters to match your business requirements.	From the Control Center, click with the right mouse button on the database you want to tune and select **Configure performance**.
Backup Database	Determine, create, and schedule a backup plan.	From the Control Center, click with the right mouse button on the database you want to backup and select **Backup**->**Database using SmartGuide**.
Restore Database	Recover a database after a failure. It helps you understand which backup to use, and which logs to replay.	From the Control Center, click with the right mouse button on the database you want to restore and select **Restore**->**Database using SmartGuide**.
Create Table	Select basic data types, and create a primary key for the table.	From the Control Center, click with the right mouse button on the **Tables** icon and select **Create**->**Table using SmartGuide**.
Create Table Space	Create a new table space.	From the Control Center, click with the right mouse button on the **Table spaces** icon and select **Create**->**Table space using SmartGuide**.

Online Help

Online help is available with all DB2 components. The following table describes the various types of help.

Type of Help	Contents	How to Access...
Command Help	Explains the syntax of commands in the command line processor.	From the command line processor in interactive mode, enter: **?** *command* where *command* is a keyword or the entire command. For example, **?** *catalog* displays help for all the CATALOG commands, whereas **?** *catalog database* displays help for the CATALOG DATABASE command.
Control Center Help	Explains the tasks you can perform in a window or notebook. The help includes prerequisite information you need to know, and describes how to use the window or notebook controls.	From a window or notebook, click on the **Help** push button or press the F1 key.
Message Help	Describes the cause of a message number, and any action you should take.	From the command line processor in interactive mode, enter: **?** *message number* where *message number* is a valid message number. For example, **?** *SQL30081* displays help about the SQL30081 message. To view message help one screen at a time, enter: **?** *XXXnnnnn* **| more** where *XXX* is the message prefix, such as SQL, and *nnnnn* is the message number, such as 30081. To save message help in a file, enter: **?** *XXXnnnnn* > *filename.ext* where *filename.ext* is the file where you want to save the message help. **Note:** On UNIX-based systems, enter: **\?** *XXXnnnnn* **| more** or **\?** *XXXnnnnn* > *filename.ext*

Type of Help	**Contents**	**How to Access...**
SQL Help	Explains the syntax of SQL statements.	From the command line processor in interactive mode, enter: **help** *statement* where *statement* is an SQL statement. For example, **help** *SELECT* displays help about the SELECT statement.
SQLSTATE Help	Explains SQL states and class codes.	From the command line processor in interactive mode, enter: **?** *sqlstate* or **?** *class-code* where *sqlstate* is a valid five digit SQL state and *class-code* is a valid two digit class code. For example, **?** *08003* displays help for the 08003 SQL state, whereas **?** *08* displays help for the 08 class code.

DB2 Books

The table in this section lists the DB2 books. They are divided into two groups:

- Cross-platform books: These books are for DB2 on any of the supported platforms.
- Platform-specific books: These books are for DB2 on a specific platform. For example, there is a separate *Quick Beginnings* book for DB2 on OS/2, Windows NT, and UNIX-based operating systems.

Most books are available in HTML and PostScript format, and in hardcopy that you can order from IBM. The exceptions are noted in the table.

You can obtain DB2 books and access information in a variety of different ways:

View To view an HTML book, you can do the following:

- If you are running DB2 administration tools on OS/2, Windows 95, or the Windows NT operating systems, you can use the Information Center. “About the Information Center” on page 350 has more details.
- Use the open file function of the Web browser supplied by DB2 (or one of your own) to open the following page:

  sqllib/doc/html/index.htm

  The page contains descriptions of and links to the DB2 books. The path is located on the drive where DB2 is installed.

  You can also open the page by double-clicking on the **DB2 Online Books** icon. Depending on the system you are using, the icon is in the main product folder or the Windows Start menu.

Search To search for information in the HTML books, you can do the following:

- Click on **Search the DB2 Books** at the bottom of any page in the HTML books. Use the search form to find a specific topic.
- Click on **Index** at the bottom of any page in an HTML book. Use the Index to find a specific topic in the book.
- Display the Table of Contents or Index of the HTML book, and then use the find function of the Web browser to find a specific topic in the book.
- Use the bookmark function of the Web browser to quickly return to a specific topic.
- Use the search function of the Information Center to find specific topics. “About the Information Center” on page 350 has more details.

Print To print a book on a PostScript printer, look for the file name shown in the table.

Order To order a hardcopy book from IBM, use the form number.

Book Name	Book Description	Form Number File Name
Cross-Platform Books		
Administration Getting Started	Introduces basic DB2 database administration concepts and tasks, and walks you through the primary administrative tasks.	S10J-8154-00 db2k0x50
Administration Guide	Contains information required to design, implement, and maintain a database to be accessed either locally or in a client/server environment.	S10J-8157-00 db2d0x50
API Reference	Describes the DB2 application programming interfaces (APIs) and data structures you can use to manage your databases. Explains how to call APIs from your applications.	S10J-8167-00 db2b0x50
CLI Guide and Reference	Explains how to develop applications that access DB2 databases using the DB2 Call Level Interface, a callable SQL interface that is compatible with the Microsoft ODBC specification.	S10J-8159-00 db2l0x50
Command Reference	Explains how to use the command line processor, and describes the DB2 commands you can use to manage your database.	S10J-8166-00 db2n0x50
DB2 Connect Enterprise Edition Quick Beginnings	Provides planning, installing, configuring, and using information for DB2 Connect Enterprise Edition. Also contains installation and setup information for all supported clients.	S10J-7888 db2cyx50
DB2 Connect Personal Edition for Windows and OS/2 Enviroments Quick Beginnings	Provides planning, installing, configuring, and using information for DB2 Connect Personal Edition.	S10J-8162-00 db2c1x50
DB2 Connect User's Guide	Provides concepts, programming and general using information about the DB2 Connect products.	S10J-8163-00 db2c0x50
DB2 Connectivity Supplement	Provides setup and reference information for customers who want to use DB2 for AS/400, DB2 for OS/390, DB2 for MVS, or DB2 for VM as DRDA Application Requesters with Database Server servers, and customers who want to use DRDA Application Servers with DB2 Connect (formerly DDCS) application requesters. **Note:** Available in HTML and PostScript formats only.	No form number db2h1x50
Embedded SQL Programming Guide	Explains how to develop applications that access DB2 databases using embedded SQL, and includes discussions about programming techniques and performance considerations.	S10J-8158-00 db2a0x50
Glossary	Provides a comprehensive list of all DB2 terms and definitions. **Note:** Available in HTML format only.	No form number db2t0x50

Book Name	Book Description	Form Number File Name
Installing and Configuring DB2 Clients	Provides installation and setup information for all DB2 Client Application Enablers and DB2 Software Developer's Kits. **Note:** Available in HTML and PostScript formats only.	No form number db2iyx50
Master Index	Contains a cross reference to the major topics covered in the DB2 library. **Note:** Available in PostScript format and hardcopy only.	S10J-8170-00 db2w0x50
Message Reference	Lists messages and codes issued by DB2, and describes the actions you should take.	S10J-8168-00 db2m0x50
IBM Replication Guide and Reference	Provides planning, configuring, administering, and using information for the IBM Replication tools supplied with DB2.	S95H-0999 db2e0x50
Roadmap to DB2 Programming	Introduces the different ways your applications can access DB2, describes key DB2 features you can use in your applications, and points to detailed sources of information for DB2 programming.	S10J-8155-00 db2u0x50
SQL Getting Started	Introduces SQL concepts, and provides examples for many constructs and tasks.	S10J-8156-00 db2y0x50
SQL Reference	Describes SQL syntax, semantics, and the rules of the language. Also includes information about release-to-release incompatibilities, product limits, and catalog views.	S10J-8165-00 db2s0x50
System Monitor Guide and Reference	Describes how to collect different kinds of information about your database and the database manager. Explains how you can use the information to understand database activity, improve performance, and determine the cause of problems.	S10J-8164-00 db2f0x50
Troubleshooting Guide	Helps you determine the source of errors, recover from problems, and use diagnostic tools in consultation with DB2 Customer Service.	S10J-8169-00 db2p0x50
What's New	Describes the new features, functions, and enhancements in Database Server. **Note:** Available in HTML and PostScript formats only.	No form number db2q0x50
	Platform-Specific Books	
Building UNIX Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a UNIX system.	S10J-8161-00 db2axx50
Building OS/2 and Windows Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Windows or OS/2 system.	S10J-8160-00 db2a1x50

Book Name	Book Description	Form Number File Name
DB2 Extended Enterprise Edition Quick Beginnings	Provides planning, installing, configuring, and using information for Database Server Extended Enterprise Edition for AIX.	S72H-9620 db2v3x50
DB2 Personal Edition Quick Beginnings	Provides planning, installing, configuring, and using information for Database Server Personal Edition on OS/2, Windows 95, and the Windows NT operating systems.	S10J-8150 db2i1x50
DB2 SDK for Macintosh Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Macintosh system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S50H-0528 sqla7x02
DB2 SDK for SCO OpenServer Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SCO OpenServer system. **Note:** Available for DB2 Version 2.1.2 only.	S89H-3242 sqla9x02
DB2 SDK for Silicon Graphics IRIX Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a Silicon Graphics system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S89H-4032 sqlaax02
DB2 SDK for SINIX Building Your Applications	Provides environment setup information and step-by-step instructions to compile, link, and run DB2 applications on a SINIX system. **Note:** Available in PostScript format and hardcopy for DB2 Version 2.1.2 only.	S50H-0530 sqla8x00
Quick Beginnings for OS/2	Provides planning, installing, configuring, and using information for Database Server on OS/2. Also contains installing and setup information for all supported clients.	S10J-8147 db2i2x50
Quick Beginnings for UNIX	Provides planning, installing, configuring, and using information for Database Server on UNIX-based platforms. Also contains installing and setup information for all supported clients.	S10J-8148 db2ixx50
Quick Beginnings for Windows NT	Provides planning, installing, configuring, and using information for Database Server on the Windows NT operating system. Also contains installing and setup information for all supported clients.	S10J-8149 db2i6x50

Notes:

1. The character in the sixth position of the file name indicates the language of a book. For example, the file name `db2d0e50` indicates that the *Administration Guide* is in English. The following letters are used in the file names to indicate the language of a book:

Language	Identifier	Language	Identifier
Brazilian Portuguese	B	Hungarian	H
Bulgarian	U	Italian	I
Czech	X	Norwegian	N
Danish	D	Polish	P
English	E	Russian	R
Finnish	Y	Slovenian	L
French	F	Spanish	Z
German	G	Swedish	S

2. For late breaking information that could not be included in the DB2 books, see the README file. Each DB2 product includes a README file which you can find in the directory where the product is installed.

About the Information Center

The Information Center provides quick access to DB2 product information. The Information Center is available on OS/2, Windows 95, and the Windows NT operating systems. You must install the DB2 administration tools to see the Information Center.

Depending on your system, you can access the Information Center from the:

- Main product folder
- Toolbar in the Control Center
- Windows Start menu.

The Information Center provides the following kinds of information. Click on the appropriate tab to look at the information:

Tasks Lists tasks you can perform using DB2.

Reference Lists DB2 reference information, such as keywords, commands, and APIs.

Books Lists DB2 books.

Troubleshooting Lists categories of error messages and their recovery actions.

Sample Programs Lists sample programs that come with the DB2 Software Developer's Kit. If the Software Developer's Kit is not installed, this tab is not displayed.

Web Lists DB2 information on the World Wide Web. To access this information, you must have a connection to the Web from your system.

When you select an item in one of the lists, the Information Center launches a viewer to display the information. The viewer might be the system help viewer, an editor, or a Web browser, depending on the kind of information you select.

The Information Center provides search capabilities so you can look for specific topics, and filter capabilities to limit the scope of your searches.

Appendix B. What's New in IBM Replication

The DataPropagator Relational Version 1 (DPROPR V1) products have been updated for Version 5 (V5) and are now known collectively as IBM Replication. IBM Replication V5 provides the replication capabilities you learned to expect from DPROPR V1 plus many new features and integration with DB2 Universal Database V5.

Packaging Integration

Most Capture and Apply programs are now packaged with the database.

- IBM Replication for Windows NT, Windows '95, OS/2, AIX, Solaris, and HP-UX are packaged with DB2 Universal Database V5.1.
- DPROPR Capture for VSE and VM V5.1 are packaged with DB2 Server for VSE and VM which includes DB2 Universal Database for administration.
- DPROPR Capture and Apply for MVS V5.1 are separately available .

Improved Administration

IBM Replication provides a higher level of integration with DB2 on your client administration platforms.

- A lower cost of administration:
  - Administration for IBM Replication is installed with the DB2 Universal Database on the Windows NT, Windows 95, and OS/2 platforms and is handled through the Control Center, the administration tool of the DB2 Universal Database.
  - Administration though the Control Center provides features for power users to define multiple registration, now known as “defining replication sources,” or multiple subscriptions, in a single user action.
- Replication administration now supports both "push" and "pull" configurations; these configurations allow the Apply program to push data to or pull data from the target table:
  - Push configurations are used for low latency requirements
  - Pull configurations provide better performance
  - Push and pull configurations enable update-anywhere and mobile replication
- IBM Replication administration now supports view replication sources and subscriptions:
  - You can define an n-way join view as a replication source in a single action.
  - You can subscribe to an n-way join replication source as a single action.
- The Control Center provides flexible customization of the replication environment for your site's requirements.

Batch Application Stream Integration

IBM Replication V5 provides better integration with your batch application stream.

- Distributed event management:

  Your applications can drive subscriptions locally, or across an entire network.

- End-of-period controls:

  You can specify not just *when* replication should occur, but *which* changes should be replicated, enabling you to coordinate replication with your business practices, such as daily accounting periods.

New Target Table Types

New target table types have been added to fit your application needs.

- User copy tables:

  Read-only copies, without any additional timestamp or control columns

- Replica tables:

  Horizontal fragments of your existing application tables, which are all updateable anywhere and anytime.

Data Consistency

IBM Replication supports group subscriptions and replication for logical partitioning keys.

- Replication subscriptions have multiple members that can be grouped for referential integrity. All the members in the replication subscription are replicated in the same transaction.
- Support for logical partitioning keys provides independence from key updates. Subscription predicates are now as easy for updated columns as for non-updated columns. The partitioning of copy table spaces is independent of source table partitioning.

Improved Reliability

IBM Replication has improved reliability:

- Better resource handling for batch and recovery scenario
- Planned replication thresholds, achieving commit thresholds
- Automatic thresholds when secondary resource-limited conditions arise

Improved Performance for the Capture and Apply Programs

The Capture and Apply programs capture and replicate data faster and more efficiently.

- Reduced Apply program connect processing:

Grouped subscriptions reduce the number of times the Apply program connects to the source, control, and target servers. For example, when maintaining 40 copies, a group replication subscription replaces 40 individual subscriptions. Connect processing is reduced 40:1.

- Apply program notification:

  The Capture program now notifies the Apply program when there are changes to be applied to avoid unnecessary dynamic SQL and joins at the source site.

- Non-blocking protocol:

  - The Apply program uses a new non-blocking protocol with the Capture program, taking advantage of *uncommitted read* isolation to avoid contention and allows for a 100% Capture duty cycle, regardless of the Apply program's activity.
  - The protocol with the Apply program is nonlocking, allowing you to choose the Capture program commit interval independently of the DB2 lock timeout value. The Apply program processes do not timeout if the Capture program commit interval is increased, which reduces the PREPARE and COMMIT costs of change capture.

- Static SQL performance for the Capture program:

  The Capture program benefits from changes in DB2 for MVS Version 5.1, which allow dynamic SQL statements to be cached in the EDM pool, just as static SQL statements are cached. These changes effectively let the Capture program, a dynamic SQL application, run with static SQL performance.

Operational and Maintenance Enhancements

IBM Replication has made it easier for you to operate the replication tools and maintain the control information.

- Maintaining the Apply program is simpler:
  - Common subscription control tables reduce the effort for back up and recovery of definitions.
  - A single database can be defined as the location of all subscription control information for the entire replication network.
- You can start Capture and Apply for Windows NT in the background using Windows NT services.
- You can now specify the database on the start commands for the Capture and Apply programs, allowing automated startup of these tools for multiple databases.

Expanded Support for Run-time Processing Statements

You can run SQL statements and stored procedures at the beginning or end of each subscription cycle to manipulate data applied to your copies at both the source and target servers.

Simplified Authorization and Security

IBM Replication has a simpler administration structure than DPROPR V1, providing lower authorization costs and improved end-user authentication.

- An Apply program instance can now be run by more than one user ID, and DB2 for MVS RACF groups are supported.
- Password management for end-user authentication is more flexible; an Apply program instance can use different passwords when connecting to more than one server.

New Update-Anywhere Capability

IBM Replication now provides update-anywhere replication, enabling you to synchronize distributed databases. Update-anywhere replication provides the following support:

- Rigorous conflict detection mechanisms for declarative constraint conflicts and trigger-detected conflicts.
- Automatic compensation of conflicting transactions.
- Asynchronous transaction coordination:

  Conflicting transactions are compensated much like the familiar DB2 rollback processing. Any detected dependent transactions are also rejected and compensated.

  Conflicting transactions are marked in the Capture program control information so you can easily determine which transactions have been rejected.

Support for Occasionally-Connected Systems

IBM Replication has enhanced the Capture and Apply programs to support mobile and occasionally connected systems.

- IBM Replication provides dial and disconnect exits for telephone connectivity.
- The Apply program can defer reporting the subscription progress to the pruning control table. Deferred reporting reduces remote SQL connect processing for a replication subscription to a single connect call and optimizes remote thread allocation and password verification.
- With enhanced support for push and pull configurations, mobile users need one Apply program on their laptop computer to replicate to and from the central computer and connecting as a mobile client for update-anywhere replication. The single Apply program process on the mobile client handles posting of updates replicating to or from the laptop. While disconnected, the mobile client imposes no burden on the network because central servers are not kept busy continuously looking for the mobile client. Rather, the laptop computer “finds” the central server when it dials in.

Appendix C. Notices

Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent product, program or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter 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 the

IBM Director of Licensing,
IBM Corporation,
500 Columbus Avenue,
Thornwood, NY, 10594
USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited
Department 071
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

This publication 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.

Programming Interface Information

This book is intended to help the customer use DataPropagator Relational.

This book also documents General-use Programming Interface and Associated Guidance Information, Product-sensitive Programming Interface and Associated Guidance Information, and Diagnosis, Modification, or Tuning Information provided by DataPropagator Relational.

General-use programming interfaces allow the customer to write programs that obtain the services of DataPropagator Relational.

General-use Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

General-use programming interface

General-use Programming Interface and Associated Guidance Information...

End of General-use programming interface

Product-sensitive programming interfaces allow the customer installation to perform tasks such as diagnosing, modifying, monitoring, repairing, tailoring, or tuning of DataPropagator Relational. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM software product. Product-sensitive programming interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces might need to be changed in order to run with new product releases or versions, or as a result of service.

Product-sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Product-sensitive programming interface

Product-sensitive Programming Interface and Associated Guidance Information...

End of Product-sensitive programming interface

Diagnosis, Modification, or Tuning Information is provided to help the customer to diagnose, modify, monitor, or tune DataPropagator Relational.

Attention:

Do not use this Diagnosis, Modification, or Tuning Information as a programming interface.

Diagnosis, Modification, or Tuning Information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Diagnosis, Modification or Tuning information

Diagnosis, Modification, or Tuning Information...

End of Diagnosis, Modification or Tuning information

Trademarks

The following terms are trademarks or registered trademarks of the IBM Corporation in the United States and/or other countries:

ACF/VTAM
ADSTAR
AISPO
AIX
AIXwindows
AnyNet
APPN
AS/400
CICS
C Set++
C/370
DATABASE 2
DatagLANce
DataHub
DataJoiner
DataPropagator
DataRefresher
DB2
Distributed Relational Database Architecture
DRDA
Extended Services
FFST
First Failure Support Technology
IBM
IMS
Lan Distance
MVS/ESA
MVS/XA
NetView
OS/400
OS/390
OS/2
PowerPC
QMF
RACF
RISC System/6000
SAA
SQL/DS
SQL/400
S/370
System/370
System/390
SystemView
VisualAge
VM/ESA
VSE/ESA
VTAM
WIN-OS/2

Trademarks of Other Companies

Microsoft, Windows, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Inc.

HP-UX is a trademark of Hewlett-Packard.

Solaris is a trademark of Sun Microsystems, Inc.

SINIX is a trademark of Siemens Nixdorf.

SCO is a trademark of Santa Cruz Operations.

Other company, product, or service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Index

Contacting IBM

This section lists ways you can get more information from IBM.

If you have a technical problem, please take the time to review and carry out the actions suggested by the *Troubleshooting Guide* before contacting DB2 Customer Support. Depending on the nature of your problem or concern, this guide will suggest information you can gather to help us to serve you better.

For information or to order any of the Database Server products contact an IBM representative at a local branch office or contact any authorized IBM software remarketer.

Telephone

If you live in the U.S.A., call one of the following numbers:

- 1-800-237-5511 to learn about available service options.
- 1-800-IBM-CALL (1-800-426-2255) or 1-800-3IBM-OS2 (1-800-342-6672) to order products or get general information.
- 1-800-879-2755 to order publications.

For information on how to contact IBM outside of the United States, see Appendix A of the IBM Software Support Handbook. You can access this document by selecting the "Roadmap to IBM Support" item at: http://www.ibm.com/support/.

Note that in some countries, IBM-authorized dealers should contact their dealer support structure instead of the IBM Support Center.

World Wide Web
http://www.software.ibm.com/data/
http://www.software.ibm.com/data/db2/library/

The DB2 World Wide Web pages provide current DB2 information about news, product descriptions, education schedules, and more. The DB2 Product and Service Technical Library provides access to frequently asked questions, fixes, books, and up-to-date DB2 technical information. (Note that this information may be in English only.)

Anonymous FTP Sites
ftp.software.ibm.com

Log on as anonymous. In the directory /ps/products/db2, you can find demos, fixes, information, and tools concerning DB2 and many related products.

Internet Newsgroups
comp.databases.ibm-db2, bit.listserv.db2-l

These newsgroups are available for users to discuss their experiences with DB2 products.

CompuServe
GO IBMDB2 to access the IBM DB2 Family forums

All DB2 products are supported through these forums.

To find out about the IBM Professional Certification Program for Database Server, go to http://www.software.ibm.com/data/db2/db2tech/db2cert.html

Part Number: S59H0999

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

S59H0999

S95H-0999-00

Spine information:

IBM Replication Guide and Reference *Version 5*